Wechall Challenges Writeup & 知识拓展

原创

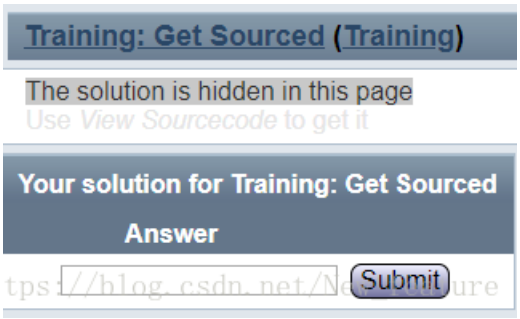
那酷小样 于 2018-08-06 17:17:44 发布 3924 收藏 5

分类专栏： 网络安全 文章标签： Wechall Writeup Wechall CTF writeup

本文链接：https://blog.csdn.net/New_feature/article/details/81452214

1 Training: Get Sourced

题目链接：http://www.wechall.net/challenge/training/get_sourced/index.php



原题：The solution is hidden in this page

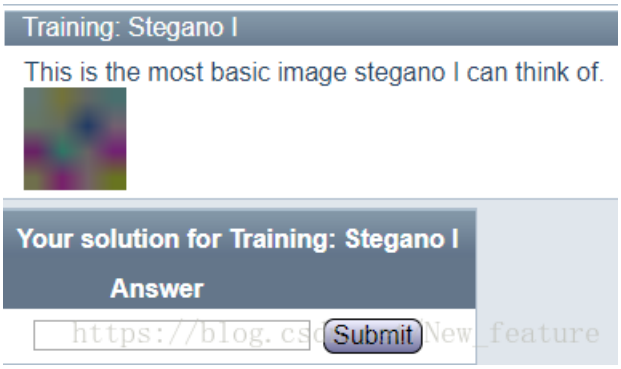
译文：解答隐藏在这个页面中

解答：在原题页面按F12或者右击查看源代码

可以看到 **You are looking for this password: html_sourcecode** ， 答案即为**html_sourcecode**

2 Training: Stegano I

题目链接：http://www.wechall.net/challenge/training/stegano1/index.php

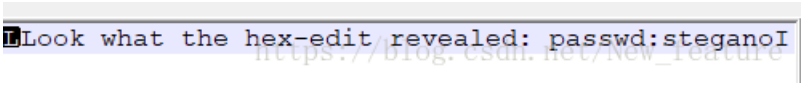


原题：This is the most basic image stegano I can think of.
译文：这是我们想象到的最基础的图片隐写

从题目中我们可以知道这是一个图片隐写，那么我们首先把图片下载下来，在图片上右击，图片另存为到桌面，再打开这个图片发现只有一个点大小，放大也什么都没有，因此不会再其中隐藏二维码，此路不通换条路。

然后我们查看图片属性，在信息中没有看到特殊的信息或字符，不存在flag，而我们看到图片大小只有102字节，对一个图片来说，太小了，不太正常。

因此查看一下图片的源码，右击记事本或notepad++等文字编辑器打开，发现前面都是乱码，最后有一条英文提示为：



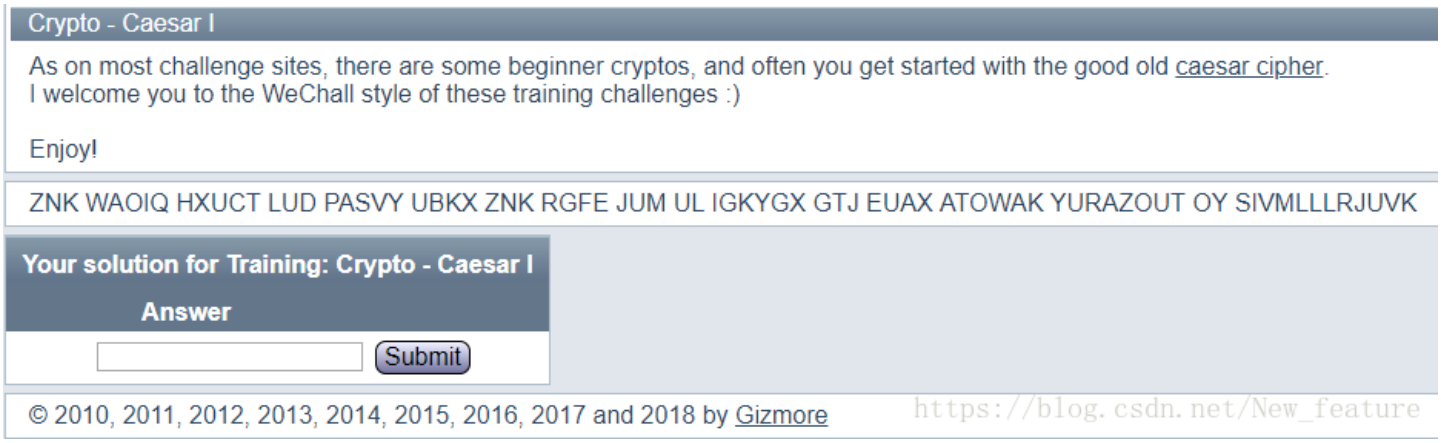
passwd:**steganol**

即找到答案为：**steganol**

图片隐写知识拓展：**待完成**

3 Training: Crypto - Caesar I

题目链接：http://www.wechall.net/challenge/training/crypto/caesar/index.php

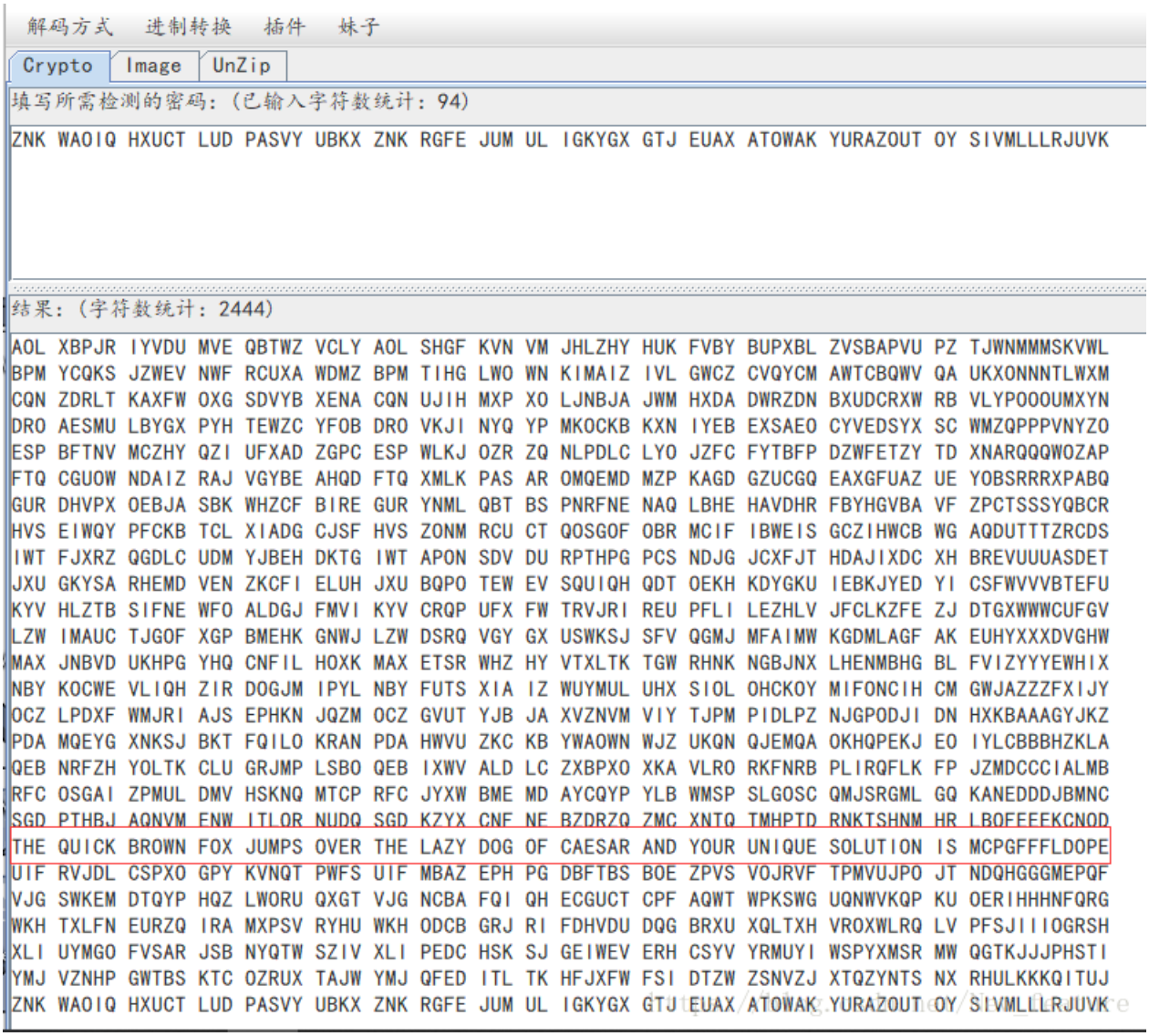


原题：As on most challenge sites, there are some beginner cryptos, and often you get started with the good old caesar cipher.I welcome you to the WeChall style of these training challenges :)Enjoy!
译文：和大多数挑战网站一样，有一些初学密码，通常开始会让你使用凯撒密码。我欢迎你接受这些培训挑战的WeChall风格：）享受！
密文为： ZNK WAOIQ HXUCT LUD PASVY UBKX ZNK RGFE JUM UL IGKYGX GTJ EUAX ATOWAK YURAZOUT OY SIVMLLLRJUVK

根据题意，我们推测该密文使用了凯撒密码

在这里我是用了CTFtools工具，进行密文的解码
工具分享：**待完成**

在”解码方式”中选择凯撒解码，看到结果有很多条，这个时候需要淡定。。。
我们对这些结果进行分析，发现方框中的这条结果貌似的有一定的含义，试着翻译一下，然而前几个单词可能不认识？
但是最后几个单词组在一起YOUR UNIQUE SOLUTION IS MCPGFFFLDOPE
翻译过来：你唯一的解决方案是MCPGFFFLDOPE
我们试着提交以下这个答案，发现，没错就是它，就是这么简单！！！

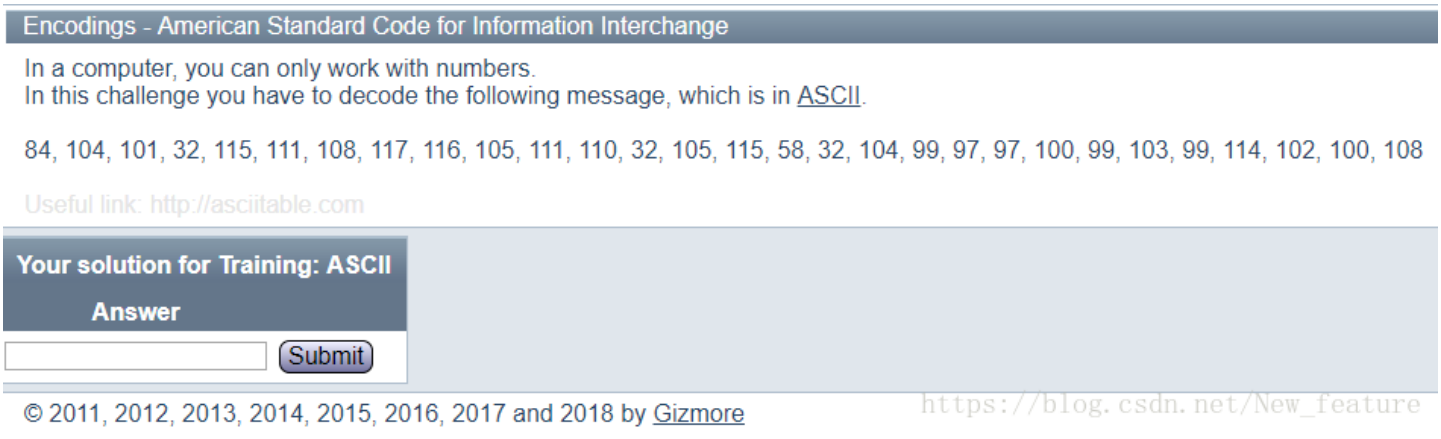


4 Training: WWW-Robots

题目链接：http://www.wechall.net/challenge/training/www/robots/index.php
略

5 Training: ASCII

题目链接：http://www.wechall.net/challenge/training/encodings/ascii/index.php

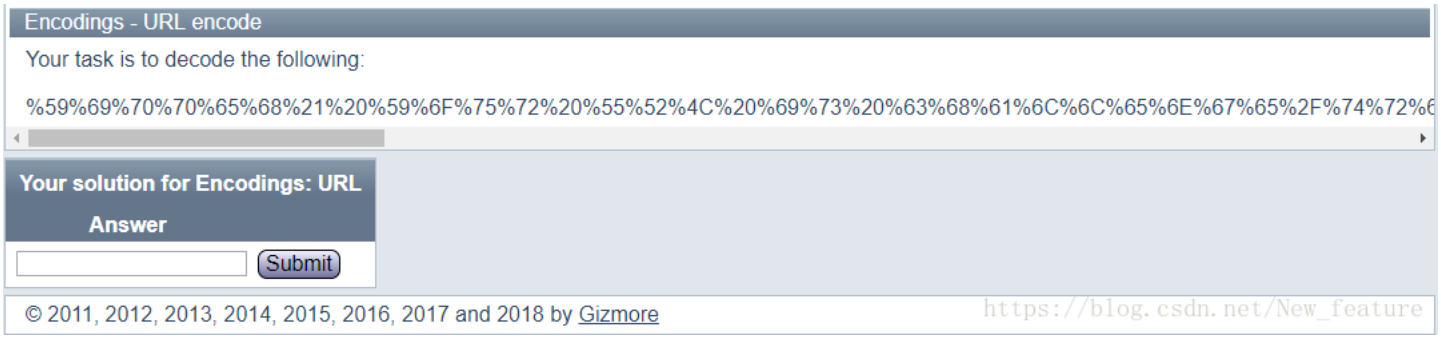


原题： In a computer, you can only work with numbers.
In this challenge you have to decode the following message, which is in ASCII.
译文： 在计算机中，您只能使用数字。
在此挑战中，您必须解码以下消息，该消息是ASCII格式。

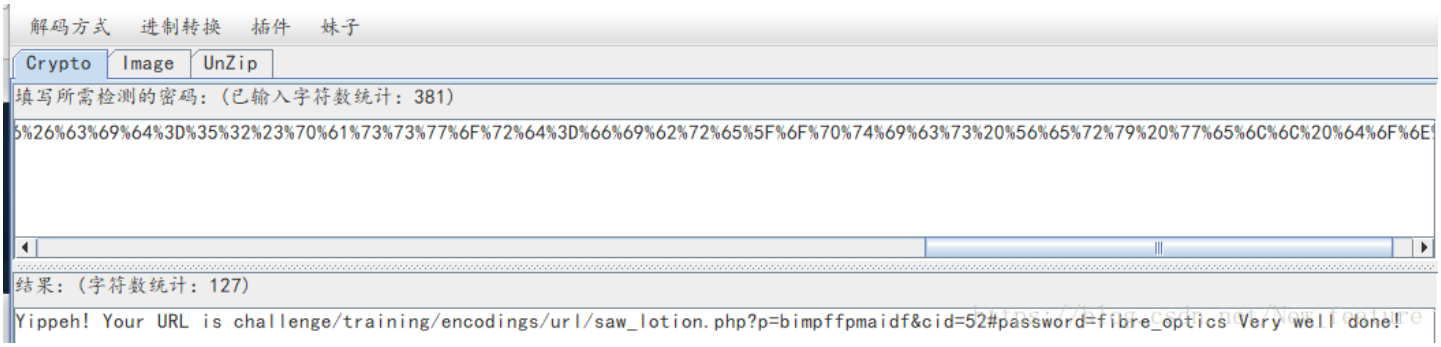
题目提示该题与ASC相关，因此，我们猜测这一串数字应该为ASC码的十进制表示，则把其转换成对应的ASC码，
利用在线的ASC码转换工具转换（也可以手动转换），即得到结果：T, h, e, , s, o, l, u, t, i, o, n, , i, s, :, , h, c, a, a, d, c, g, c, r, f, d, l
，答案为：hcaadcgcrfdl

6 Encodings: URL

题目链接：http://www.wechall.net/challenge/training/encodings/url/index.php



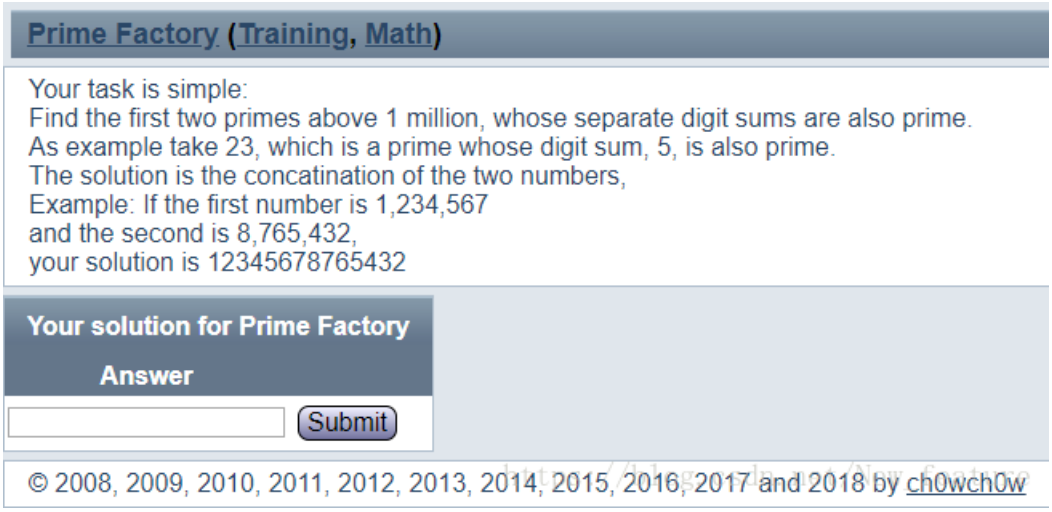
题目提示为url，我们推测使用url解码
同样使用CTFtools工具，



把这个链接添加到题目链接的后面，访问即完成本题 注意：把重复的去掉

7 Prime Factory

题目链接：http://www.wechall.net/challenge/training/prime_factory/index.php



原题：Your task is simple:
Find the first two primes above 1 million, whose separate digit sums are also prime.
As example take 23, which is a prime whose digit sum, 5, is also prime.
The solution is the concatination of the two numbers,
Example: If the first number is 1,234,567
and the second is 8,765,432,
your solution is 12345678765432

译文：你的任务很简单：
找出超过100万的前两个素数，其单独的数字总和也是素数。
例如23，这是一个素数，其数字和5也是素数。
解决方案是两个数字的连接，
示例：如果第一个数字是1,234,567
第二个是8,765,432，
您的解决方案是12345678765432

题目让我们找到大于1000000的素数，并且取前两个，很简单，让程序跑一下就可以啦

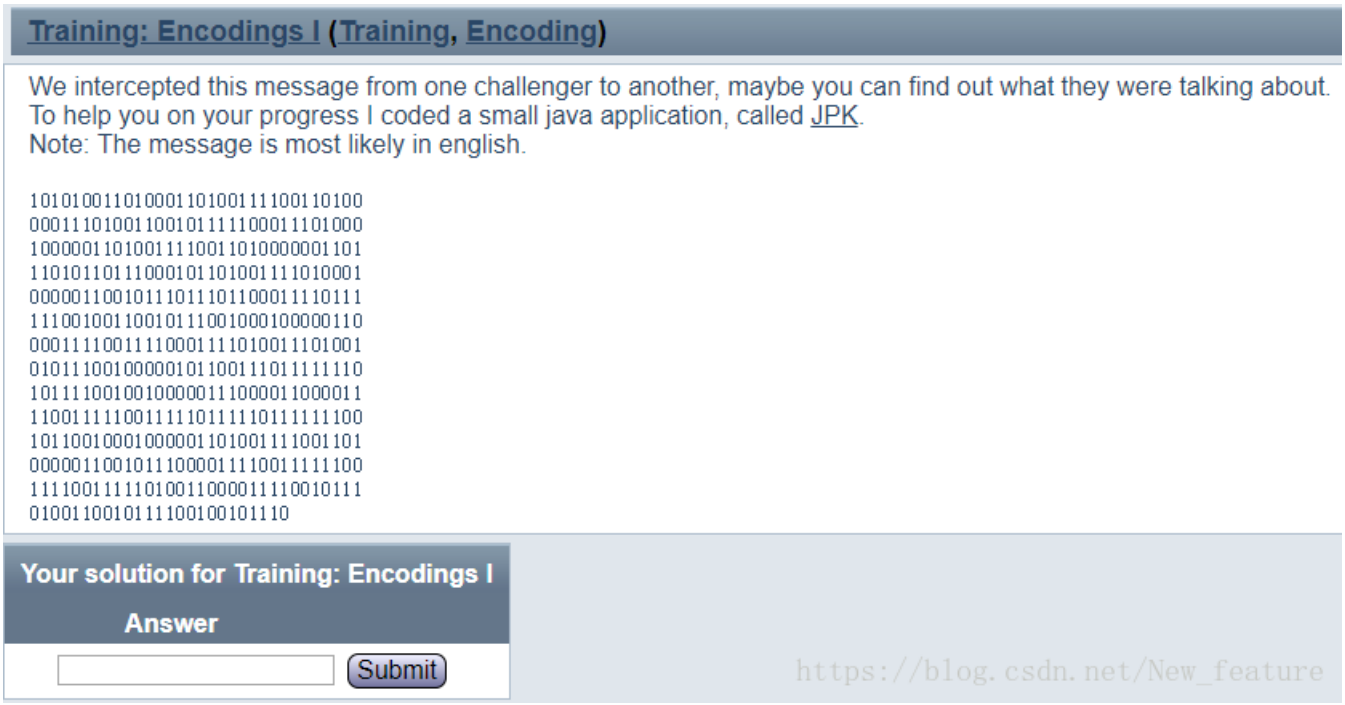
这里我就直接有python写了一个程序：

```
def fun_sum(n):           #判断一个数每位的累加和是不是素数
    sum = 0
    while n/10 != 0:
        sum = sum + n%10
        n = n // 10
    sum = sum + n
    if fun(sum) == 1 :    #是素数
        return 1
    else:
        return 0
def fun(m):               #判断一个数是不是素数
    i = 2
    for i in range(2,m-1):
        if m % i == 0:
            return 0
    return 1
if __name__ == '__main__':
    for i in range(1000000,10000000):
        if fun(i) == 1:           #是素数
            if fun_sum(i) == 1:   #也是素数
                print(i)
```

运行后得到结果，取前两个：1000033 1000037，按题目规定格式提交即可

8 Training: Encodings I

题目链接：http://www.wechall.net/challenge/training/encodings1/index.php



原题：We intercepted this message from one challenger to another, maybe you can find out what they were talking about.
To help you on your progress I coded a small java application, called JPK.
Note: The message is most likely in english.
译文：我们从一个挑战者到另一个挑战者拦截了这条消息，也许你可以找到他们在谈论的内容。
为了帮助您完成进度，我编写了一个名为JPK的小型Java应用程序。
注意：该消息很可能是英文的。

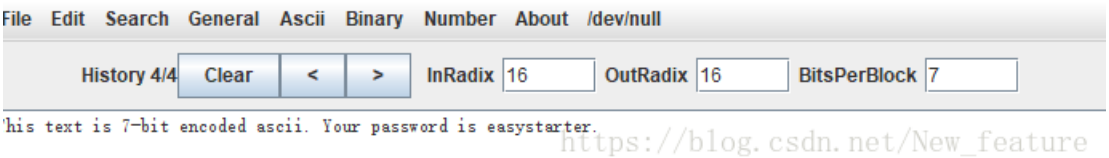
题目要求我们解析这条消息，同时还为我们提供了一个工具JPK，下载下来，观察有一个Binary（二进制）模块

把代码拷进去，然后我们要对这些数字进行拆分，我们需要知道，一个ASC码字符有7位或者8位二进制表示，首先我们在这个工具中设置宽度位8位，在选择Binary中的Binary format选项，生成7位二进制格式

再选择Binary 中的 Binary to ASC Ⅱ生成ASC码，发现无规律，是乱码

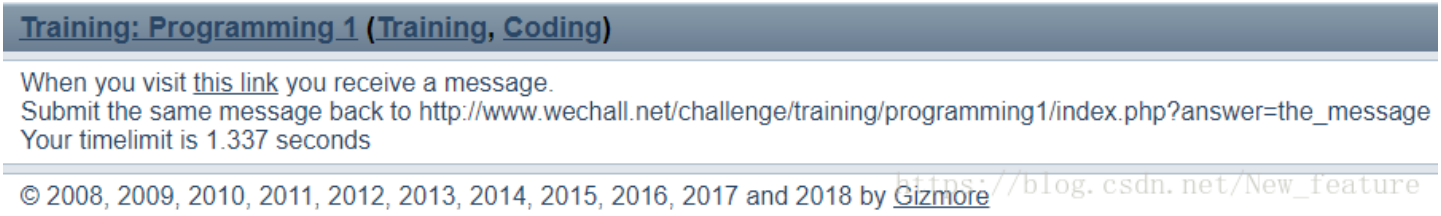
此路不通，然后撤回，设置宽度位8位，重复上面步骤。即得到答案结果
拆分：

转ASC Ⅱ码：



9 Training: Programming 1

题目链接：http://www.wechall.net/challenge/training/programming1/index.php

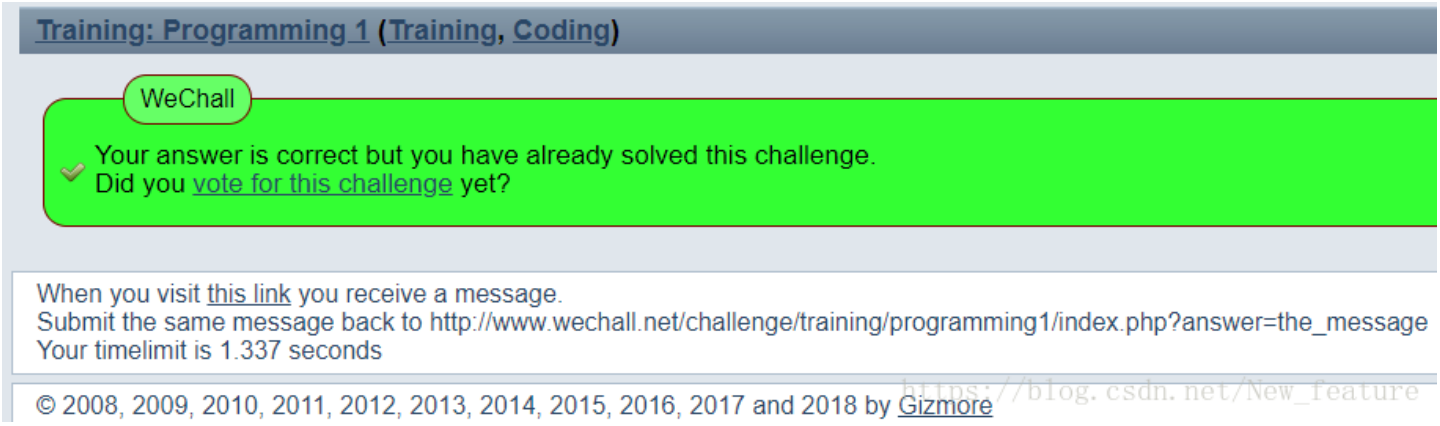


原题：When you visit **this link** you receive a message.Submit the same message back to http://www.wechall.net/challenge/training/programming1/index.php?answer=the_message，Your timelimit is 1.337 seconds
译文：当您访问**此链接**时，您会收到一条消息。将相同的消息提交回 http://www.wechall.net/challenge/training/programming1/index.php?answer=the_message 你的时间限制是1.337秒
根据题意可知，把 this link 链接里面获得的信息，加到第二个链接后面，赋值给 answer 提交。
按照正常的手动操作来执行，时间肯定超过了1.337s，那么只能跑代码

```
import urllib.request
import http.cookiejar
import webbrowser

url1 = 'http://www.wechall.net/challenge/training/programming1/index.php?action=request'
url2 = 'http://www.wechall.net/challenge/training/programming1/index.php?answer='
header = {}
req = urllib.request.Request(url1,headers=header)
req.add_header('cookie','自己的cookie')  #第一个是header里面的cookie属性，第二个位置填写自己的cookie值
text = urllib.request.urlopen(req).read().decode('utf-8')
url2 = url2 + text
webbrowser.open(url2)
```

然后就。。。

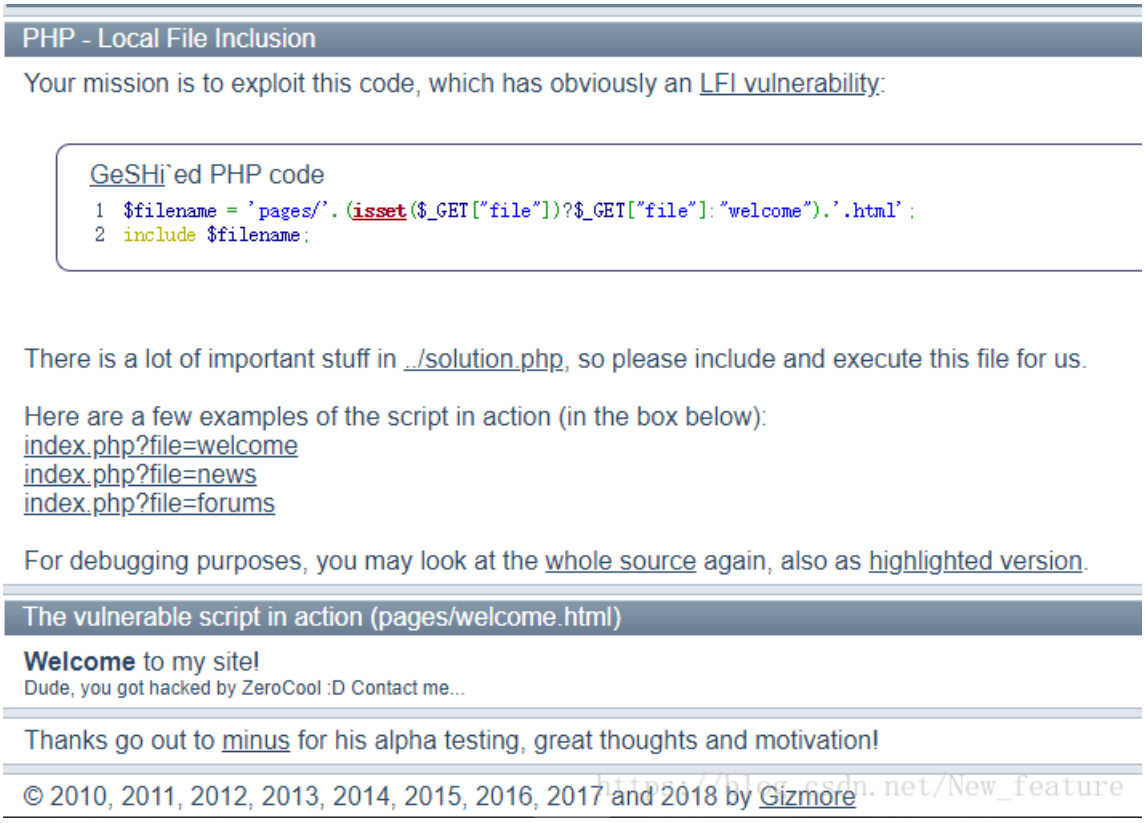


10 Training: Regex

略

11 Training: PHP LFI

题目链接：http://www.wechall.net/challenge/training/php/lfi/up/index.php



译文：您的任务是利用此代码，该代码显然存在LFI漏洞：
在../solution.php中有很多重要的东西，所以请为我们包含并执行这个文件。

以下是脚本的一些示例（在下面的框中）：
index.php?file=welcome
index.php?file=news
index.php?file=forums

出于调试目的，您可以再次查看整个源代码，也可以查看突出显示的版本。

该题目考察的是LFI，PHP的本地文件包含漏洞
分析题目中的代码，看到最后有一个.html,我们可以使用%00进行截断

既然题目提示，我们可以按照示例提交查看一下

www.wechall.net/challenge/training/php/lfi/up/index.php?file=../solution.php%00

提示：

The vulnerable script in action (pages/../solution.php.html)

PHP Warning(2): include(pages/../solution.php.html): failed to open stream: No such file or directory in ***www/challenge/training/pl***

```
Backtrace starts in www/challenge/training/php/lfi/up/index.php line 54.
eval()..... www/challenge/training/php/lfi/up/index.php(54) : eval()'d code line 1.
include()..... www/challenge/training/php/lfi/up/index.php(54) : eval()'d code line 1.
GWF_Debug::error_handler() core/inc/util/GWF_Debug.php line 183.
```

PHP Warning(2): include(): Failed opening 'pages/../solution.php.html' for inclusion (include_path='.:/usr/share/php') in ***www/challe***

```
Backtrace starts in www/challenge/training/php/lfi/up/index.php line 54.
eval()..... www/challenge/training/php/lfi/up/index.php(54) : eval()'d code line 1.
include()..... www/challenge/training/php/lfi/up/index.php(54) : eval()'d code line 1.
GWF_Debug::error_handler() core/inc/util/GWF_Debug.php line 183.
```


那么我们再往上一层目录提交看一下：

www.wechall.net/challenge/training/php/lfi/up/index.php?file=../../solution.php%00

OK 通过！！！

PHP本地文件包含漏洞 — 知识拓展： **待完成**

12 PHP 0817

题目链接：http://www.wechall.net/challenge/php0817/index.php

PHP-0817

I have written another include system for my dynamic webpages, but it seems to be vulnerable to LFI.
Here is the code:

GeSHi`ed PHP code

```php
<?php
if (isset($_GET['which']))
{
        $which = $_GET['which'];
        switch ($which)
        {
        case 0:
        case 1:
        case 2:
                require_once $which.'.php';
                break;
        default:
                echo GWF_HTML::error('PHP-0817', 'Hacker NoNoNo!', false);
                break;
        }
}
?>
```

Your mission is to include solution.php.
Here is the script in action: News, Forum, Guestbook.

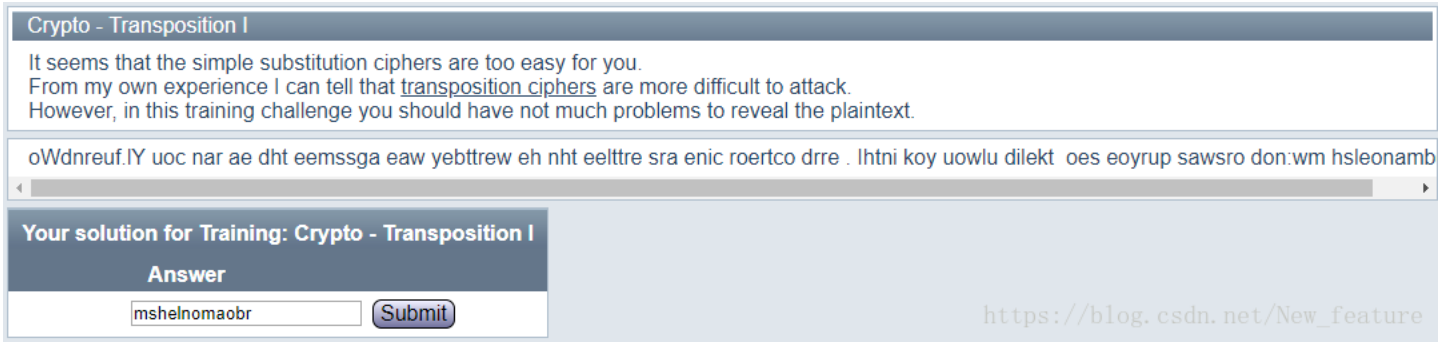
Good Luck!

这个题目与Training: PHP LFI类似，php本地文件包含漏洞
该文件solution.php
观察代码：信息赋值给which提交，并且代码后面有.php,因此只把solution赋值给which即可。

OK，通过！！！

13 Training: Crypto - Transposition I

题目链接：http://www.wechall.net/challenge/training/crypto/transposition1/index.php



译文：似乎简单的**置换密码**对你来说太容易了。
根据我自己的经验，我可以看出换位密码更难以攻击。
但是，在这次培训挑战中，您应该没有太多问题来揭示明文。

根据提示以及密文，该题应该考查**矩阵变换密码**。
首先，我们应该知道，矩阵变换密码为：把一串密文，写成矩阵的形式，然后再打乱列的顺序，即得到密文
因此，我们只需要把密文写成矩阵，根据列之间的规律关系，恢复原顺序即可。

工具：http://tholman.com/other/transposition/

我们把密文写成矩阵形式，观察规律，列数为1时没有规律，当列数位2时，发现规律，第二列在前第一列在后，可以组成一句话，该语句大概为：Wonderful. You cand the……..sword now: mshelnomaobr.
我们推测最后的答案为：mshelnomaobr。

未完待续。。。

本人纯属小白，如有不足之处，请大佬们批评指正，谢谢。