学习周记（九）暨whaleCTFweb打卡练习第一期

原创

極品一═☆宏 于 2019-02-27 09:40:16 发布 430 收藏

分类专栏： CTF_web

本文链接：https://blog.csdn.net/qq_43214809/article/details/87900745


CTF_web 专栏收录该内容

13 篇文章 0 订阅

订阅专栏

CTF_Web周练习（四）

开学后的第一次总结，新学期新气象，在下半学期里再接再厉，给自己一个好的交代。

另：攻防世界做不下去了，只好先看看别的，比如说whaleCTF_web打卡第一期（当然除sql注入外）。本writeup全部为个人答题后总结，限于本人能力有限，若有错误及不合理的地方还请各位师傅批评指正。

Keep Fighting!!! ㅅ(￣ᴗ￣)ㅅ

时间：2019年2月24日至2019年2月27日

来了，老弟！（づ￣3￣）づ ╭❤

一、writeup：

1.不明觉厉：

打开链接后是个小故事：

想都不用想直接view-source看一眼，果然没失望：

```
<!--------------------------------------------------------------------------------------------------------------------------------这个地址好特别：9s81jWjd98YU.php -->
```

直接复制到地址栏中打开，又是个登陆界面：

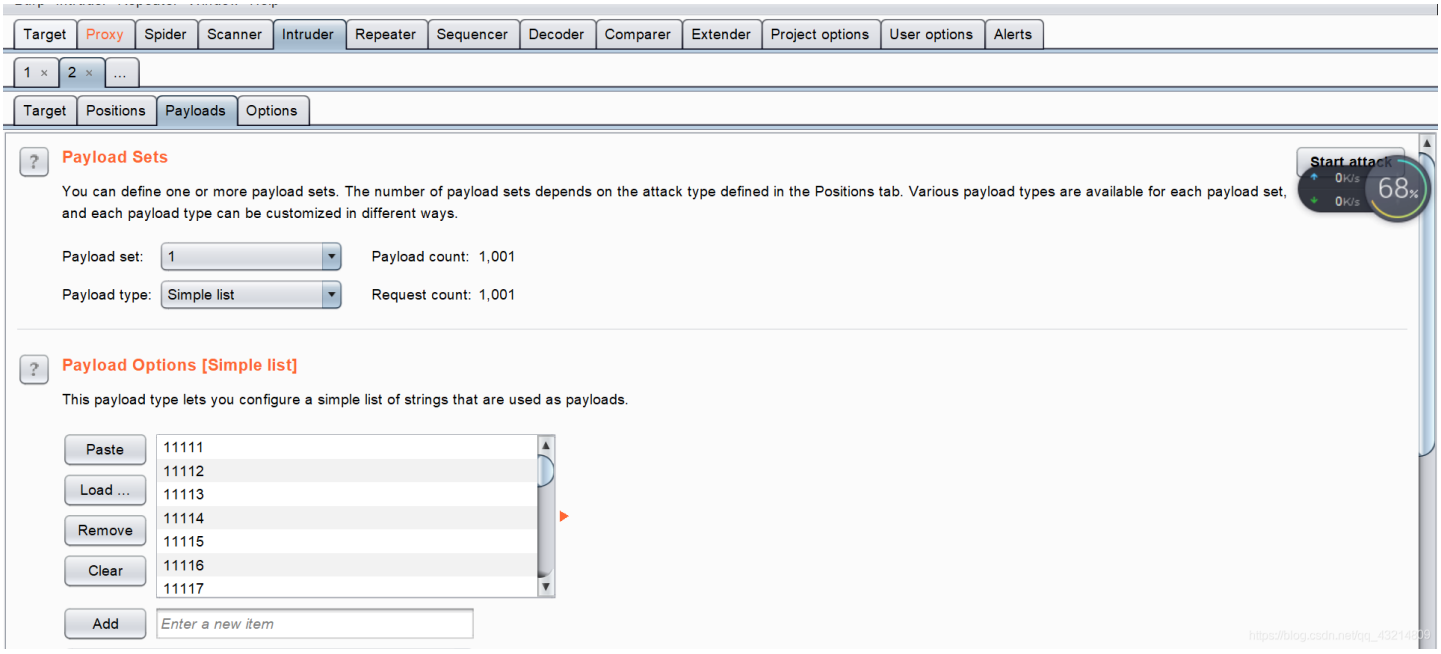
同样，view-source看一眼：

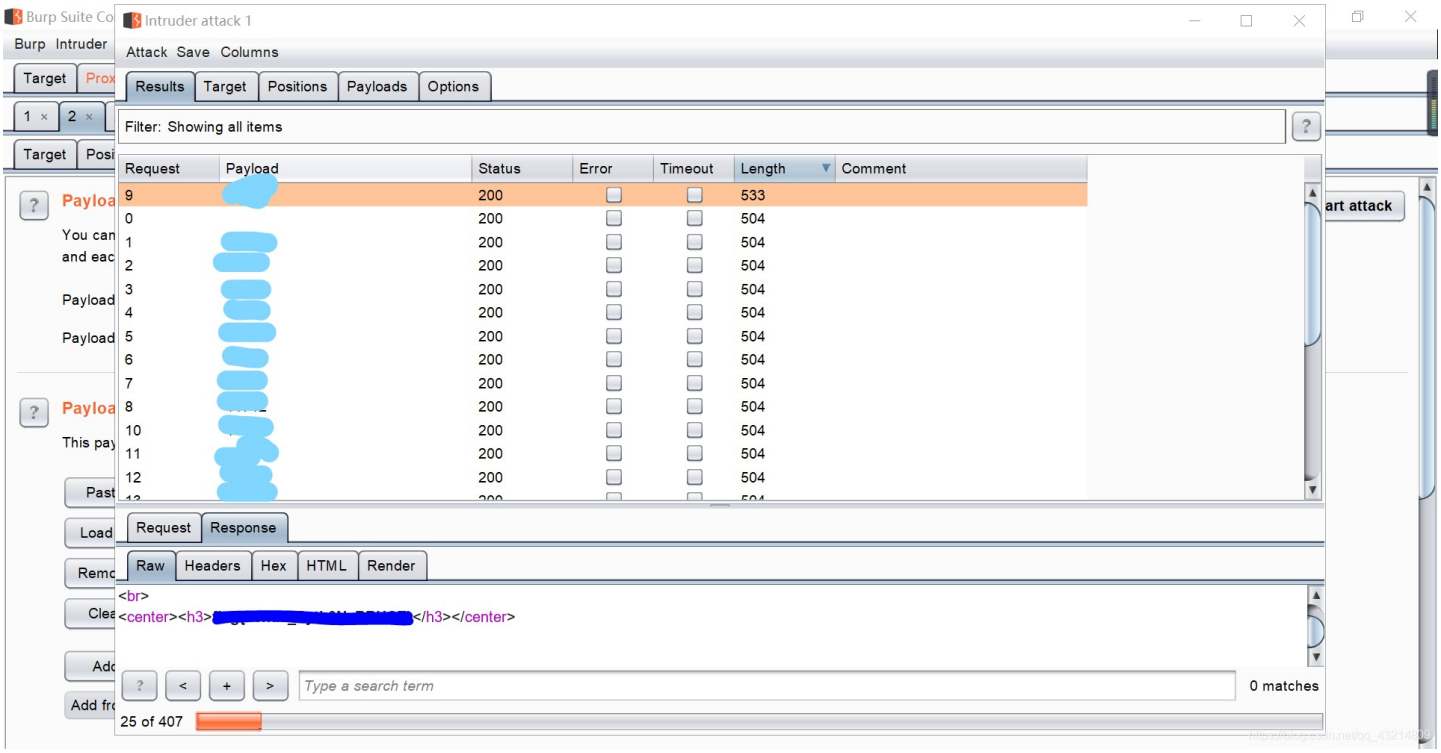
```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>9s81jWjd98YU</title>
</head>

<body>
<br>
<br>
<br />
<b>Notice</b>:  Undefined index: username in <b>/home/whaledu.com/html/web/web38/9s81jWjd98YU.php</b> on line <b>12</b><br />
<center><h3><font>注意：用户名、密码、验证码均不能为空。</font></h3></center>
<center>
    <br>
    <br>
    <br>
    <form action="9s81jWjd98YU.php" method="get">
        用户名：<input name="username" type="text" value="admin"><br><br>
        密  码：<input name="password" type="password"><br><br>
        验证码：<input name="randcode" type="text"> 016<br><br>     <input type="submit" value="确定"><br><br><br>
    </form>
</center>
</body>

<!-- 听说密码是一个数字，并且范围在(11111, 12111) -->

```

密码的范围已经给出来了，那就编写一个txt文档，从11112到12110，在之前的基本练习场中，密码泄露与这个题如出一辙，都需要用burpsuite爆破，但是这个题的难点在于验证码。验证码是随着界面的改变而刷新的，所以我们需要“控制”住验证码。在CSDN上搜了一下，一般都是自己编写一个python脚本，进行验证码爆破，看得我是一脸懵逼，后来又找了找，找到一个用burpsuite进行爆破的操作，这个确实干的漂亮，又解锁了一项新的技能。

具体的操作流程在链接里，大概就是打开Project options，在Sessions里编辑一个可以“控制”验证码的宏，然后添加到规则里去，最后抓包进行爆破。在这里，我想说的是要有耐心，我先等了两个多小时，一开始很快，后来越爆越慢，实在等不下去了，就记下个数等着第二天再来，谁知道第二天爆的时候，那个密码就差了9个，凸(艹皿艹)。

毕竟用这种方法进行爆破肯定是比较慢的，有能力的话自己写一个python脚本爆破肯定快。还有就是我一开始并没有成功，重复了好几次才弄成，先发一个空的，不填密码和验证码，进行抓包，添加规则，然后再输入密码和验证码发包后拿去爆破，这样的话就可以。

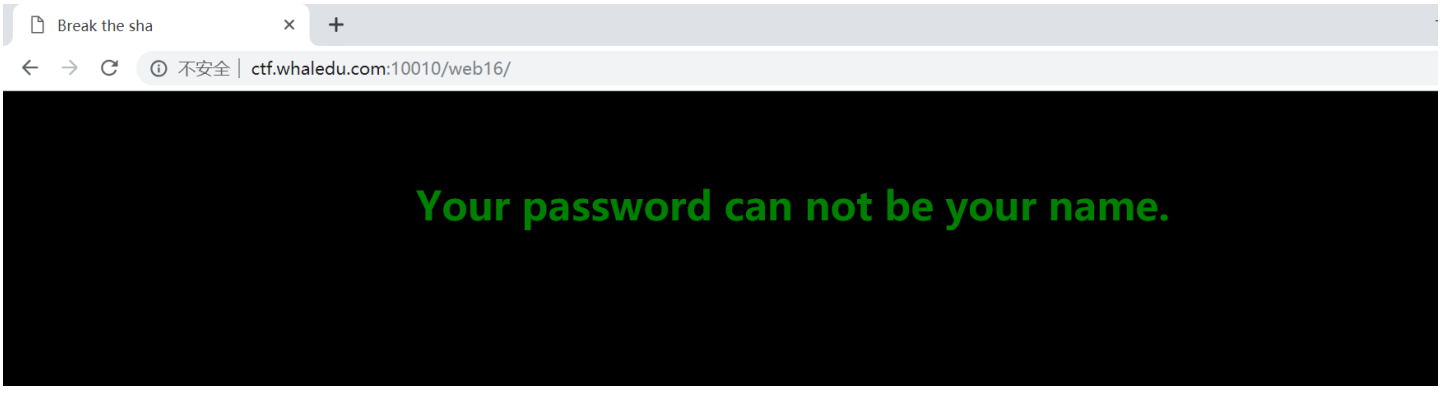




含验证码爆破链接：https://blog.csdn.net/D_pokemon/article/details/78194351

二、等量登陆：

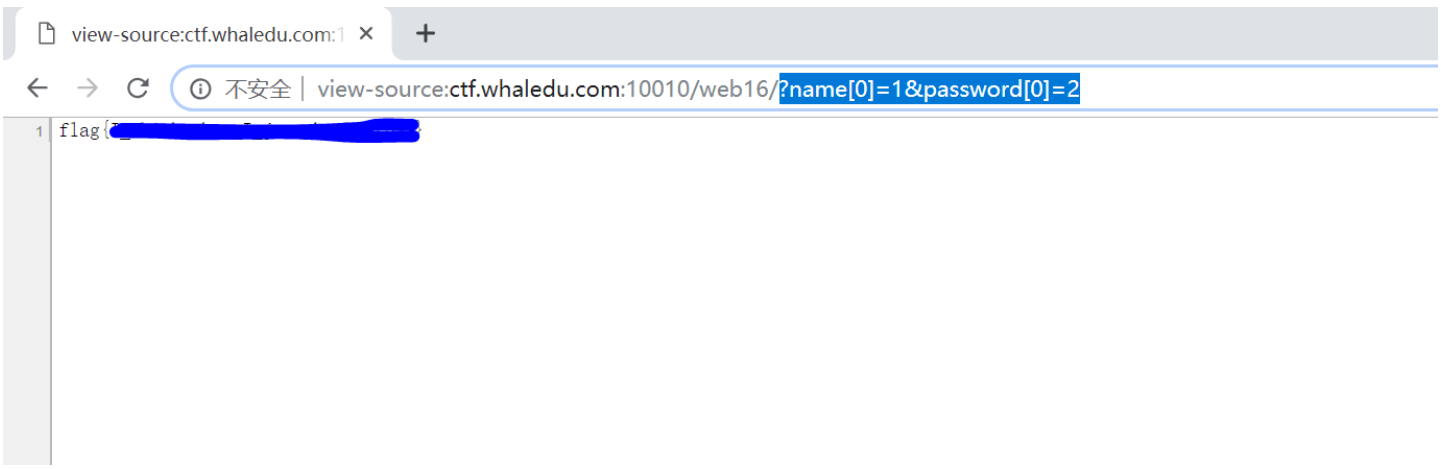
打开界面：



现在做题养成了先view-source的习惯，看了一眼，果然有东西：

提示出了index.phps，直接放到地址栏中打开，是一段php源码，开始审计：

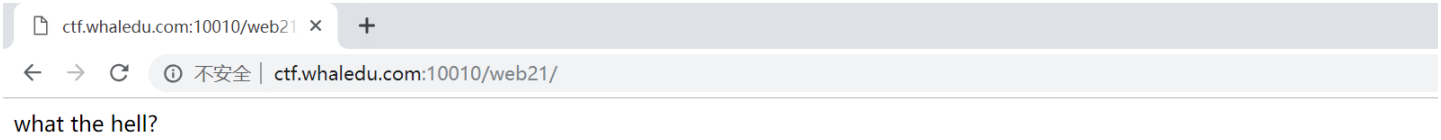
看过一遍以后，大体上知道是什么意思。第一点，我们给出的密码和用户名不能相等，这也是一开始的界面给出来的条件；第二点，它们的sha-1散列要相同。那么综合一下，这就有矛盾了，不能相等还要求散列相同。
之前也做过一道跟这个题差不多的，那道题是让原字符串和反转后的相同，但是变量不能为回文串。当时是利用了intval的边界值绕过。那么对于这道题来说，同样也需要利用一些技巧绕过。
通过搜索发现sha1()函数存在一个可绕过点，就是sha1 无法处理数组，但是php不会抛出异常，直接返回 fasle。也就是sha1([]) === false，那么这样的话，只要用户名和密码在定义时都是数组类型，那么都会返回false，这样第二个条件就可以满足了。至于第一个条件，不相同即可。这样的话直接构造payload就好：?name[0]=1&password[0]=2，直接得到flag：



后来我又查了一下，还有一个构造的方法，直接利用GET，我觉得也蛮好的，涨知识，原文是这么解释的：
如果 GET 参数中设置 name[]=a，那么 $_GET['name'] = [a]，php 会把 []=a 当成数组传入， $_GET 会自动对参数调用 urldecode。
$_POST 同样存在此漏洞，提交的表单数据，user[]=admin，_POST['user'] 得到的是 ['admin'] 是一个数组。
那么对于这个题而言可以构造：?name[]=1&password[]=2。

三、正则进入：

界面如下：

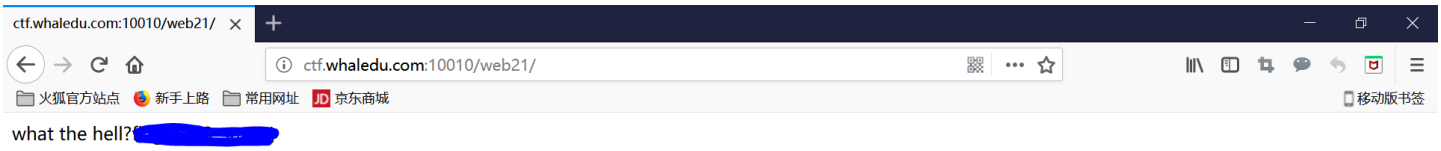


按照习惯，view-source看一眼，什么也没有，紧接着index.php看了一眼也什么都没有，然后又试了一下index.php.txt这次源码出来了：


```php
<?php
echo "waht the hell?";
$flag = "*******";
if ("POST" == $_SERVER['REQUEST_METHOD'])
{
    $password = $_POST['password'];
    if (0 >= preg_match('/^[[:graph:]]{12,}$/', $password))
    {
        echo 'Wrong Format';
        exit;
    }
    while (TRUE)
    {
        $reg = '/([[:punct:]]+|[[:digit:]]+|[[:upper:]]+|[[:lower:]]+)/';
        if (6 > preg_match_all($reg, $password, $arr))
            break;
        $c = 0;
        $ps = array('punct', 'digit', 'upper', 'lower');
        foreach ($ps as $pt)
        {
            if (preg_match("/[[:$pt:]]+/", $password))
                $c += 1;
        }
        if ($c < 3) break;
        if ("42" == $password) echo $flag;
        else echo 'Wrong password';
        exit;
    }
}
?>
```


然后开始代码审计，一看就知道与正则匹配有关，下面我们去找给出flag的条件，大的if条件先不管，先看里面的if，过掉第一个if的条件是匹配大于等于12个的可见字符串；进入到while循环后，首先是全局匹配次数大于等于6次，匹配的内容包括任何标点符号或任何数字或任何大写字母或任何小写字母。接下来，还要再进行匹配而且要使c大于等于3，也就是说给出的字符串要包括标点，数字，大小写字母中的三种及以上；最后一个就是password值为42.

综合考虑一下，要以42这个点进行构造payload，那么接下来就是想办法把符号或大小写字母加到里面去，那么首先想到能不能用00截断绕过，后来试了一下不行，应该是和要求冲突的；那么就再试一下科学记数法，因为这里面含有e，所以构造password=4.2000000e+1，即可得到flag，当然也可以选择其他的方法，不唯一：



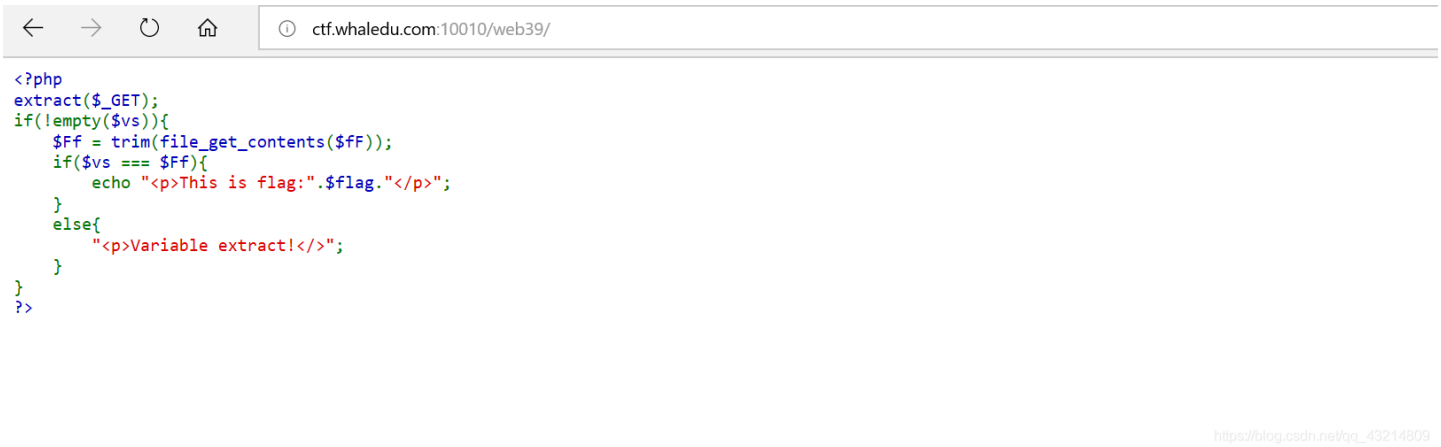
这里面涉及到的正则匹配字符簇和一些字符串如下，当作一次积累：

PHP内置字符簇：https://www.yunbook.vip/post/1545197714976.html

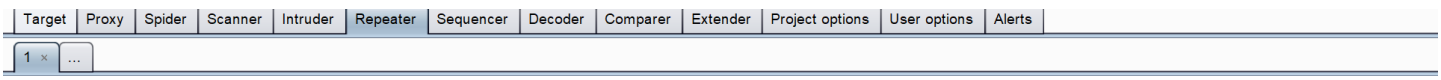
正则表达式手册：https://blog.csdn.net/qq_27905477/article/details/78819090

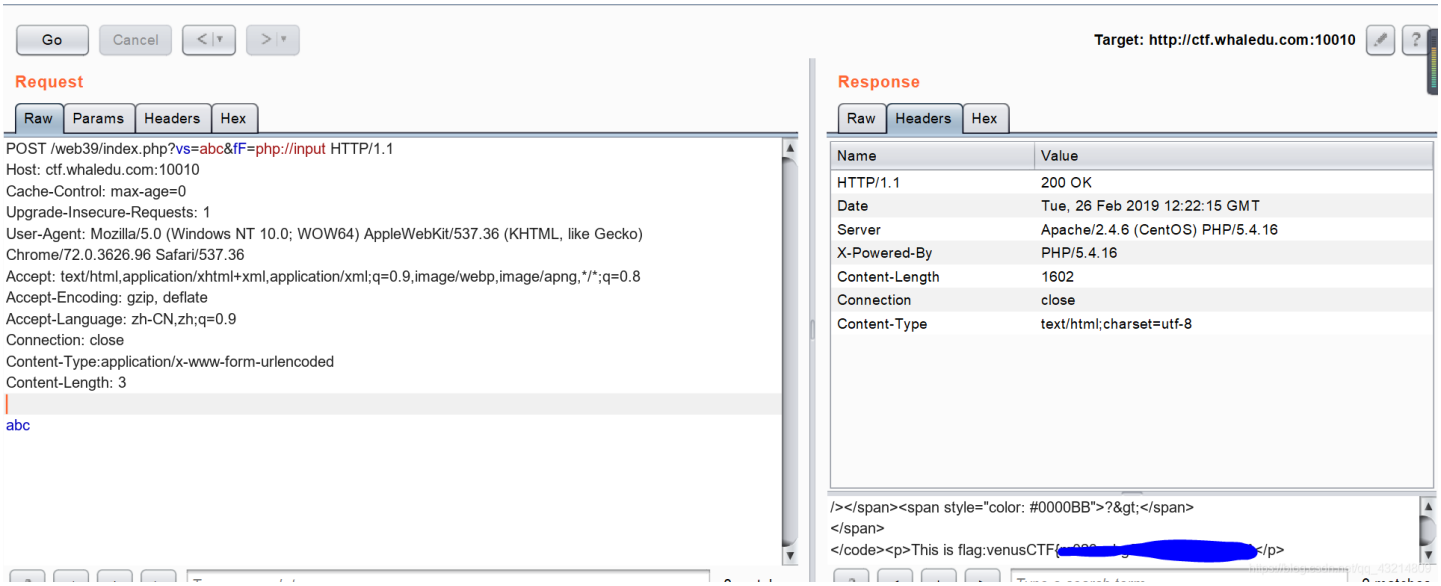
四、强势替换：

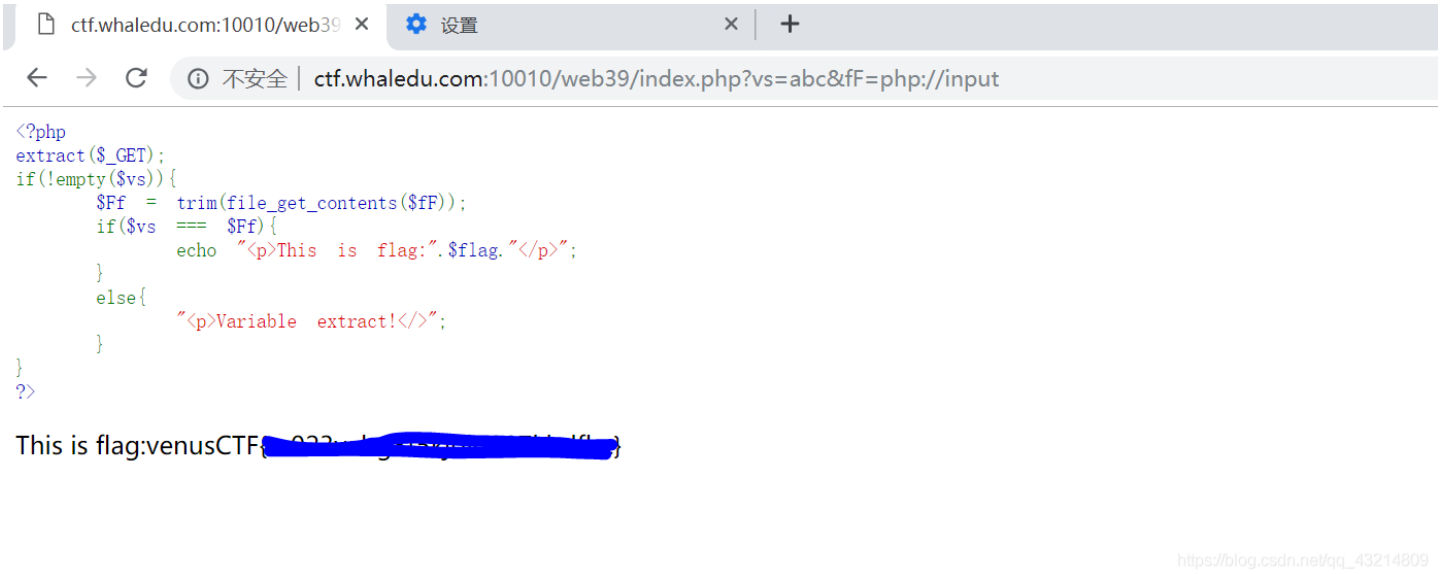
界面如下：



很简短的一段代码，extract()函数一出第一感觉全局变量覆盖，也就是说我们需要在url里添加变量vs和fF。接着往下走，首先给出了提示，vs非空，紧接着重点来了，file_get_contents()函数出来了，这就告诉我们要用到 'php://input' 来绕过，那么接下来，只要构造出payload然后POST出vs变量的值就可以，我写的很简单的payload:?vs=abc&fF=php://input，然后在Firfox的HackBar里POST 'abc'，但是很遗憾没有反应，我试了很多很多次，换了很多种payload都没有反应，后来又查了一些关于php://input的资料，该打开的也打开了，就是忘了burpsuite抓个包看看，凸(艹皿艹)真想给自己一巴掌。要是早点打开也不至于花这么长时间，抓下来后一定要添加一个Content-Type:application/x-www-form-urlencoded，因为如果是multipart/form-data，'php://input'是无效的。然后修改GET为POST，底部添加一个abc，发包即可得到flag：







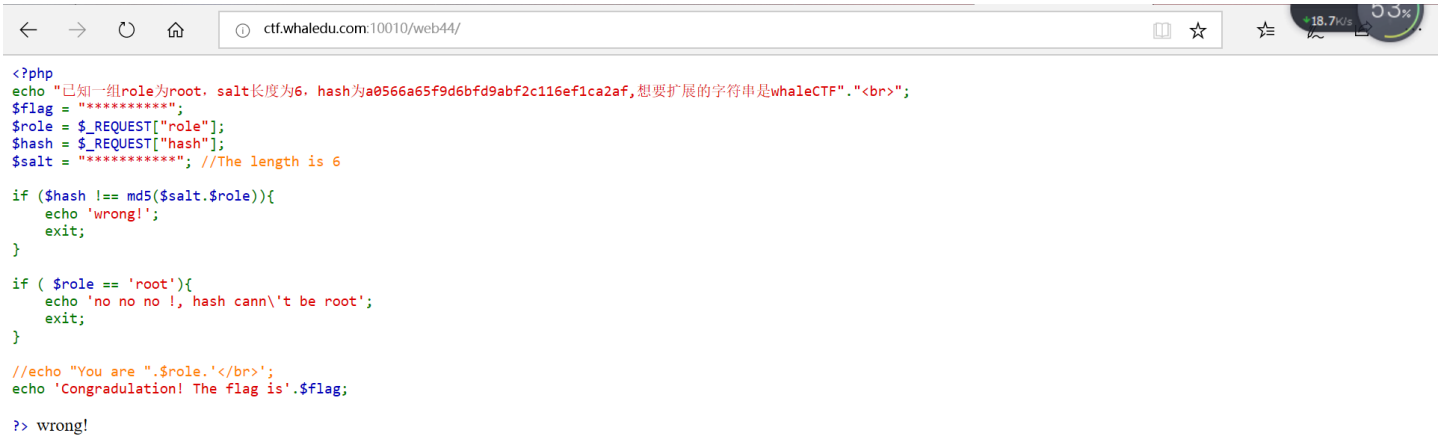
比较好的文章：

文件包含漏洞：https://www.freebuf.com/articles/web/182280.html

POST与php://input：https://blog.csdn.net/songtaiwu/article/details/79455031

五、哈希入侵：

界面如下：



这道题的考点是hash长度拓展攻击，给的很明确但是我看不懂，因为之前从来没有做过有关hash长度拓展攻击的题目，上网搜了好多资料，最后也还是不是很明白，网上有关这类题目的解析都用到了hashpump这个工具，但是我最后也没装上，就用了另外一个工具，这个工具也不错。
这道题大体上的一个思路我不是很清楚，包括补位的思想，以及相应的轮换。我只是用这个工具把flag解了出来，这一部分内容还需要后续的学习。

```
F:\python37\Web-master>python exploit.py a0566a65f9d6bfd9abf2c116ef1ca2af whaleCTF 14

               HASH_EXTENSION_ATTACK


     *       *        * * * *      * * * **
    * *     * *       *       *    *
   *   *   *   *      *        *   * * * *
  *     * *     *     *       *            *
 *       *       *    * * * *      * * * *


md5String :     a0566a65f9d6bfd9abf2c116ef1ca2af
appendString :  whaleCTF
payload :       [redacted]
newMd5String :  [redacted]

Thanks to use
```


首先的一段字符串是题目中给出的hash值，也是后面代码中的md5值，第二个whaleCTF作为拓展字符串，也就是hashpump中的Input Data to Add，最后的那个14，表示的是salt的长度6再加上'whaleCTF'的长度8.那么最后给出来的payload就是我们要的答案：

role=root%80%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00
%00%00%00%00%00%00%00%00%00
%00%00%00%00%00%00%00%00%00%00%00P%00%00%00%00%00%00%00whaleCTF&hash=aab9a3180e92bd4126d56011b672711f

```
ctf.whaledu.com:10010/web44/?role=root%80%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00

<?php
echo "已知一组role为root，salt长度为6，hash为a0566a65f9d6bfd9abf2c116ef1ca2af,想要扩展的字符串是whaleCTF"."<br>";
$flag = "**********";
$role = $_REQUEST["role"];
$hash = $_REQUEST["hash"];
$salt = "***********"; //The length is 6

if ($hash !== md5($salt.$role)){
    echo 'wrong!';
    exit;
}

if ( $role == 'root'){
    echo 'no no no !, hash cann\'t be root';
    exit;
}

//echo "You are ".$role.'</br>';
echo 'Congradulation! The flag is'.$flag;

?> Congradulation! The flag is [redacted]
```


一个很好的blog里面有工具的链接：https://blog.csdn.net/weixin_41185953/article/details/80316129

二、感受：

1.首先开学后的第一次总结，心里还是有点激动的，毕竟过了一个寒假的学习后还是能做出来点题目的，自己的能力可能并没有想象中的提高很多，但起码也在进步，进步空间还是有的；其次通过这次做题，学会了含验证码的密码爆破，对php的一些函数特性有了新的了解，正则匹配，伪协议之类的，包括hash长度拓展攻击。
要学的东西还有很多，包括这次做题还有一个sql注入的题目我没有做，一方面对于sql持敬畏心理，另一方面自己确实在sql上下的功夫还是少，还需要一段时间的学习。

2.最后日常感谢一些大哥们的知识总结，小弟受益匪浅，感谢！！！