

# 百度杯CTF比赛 九月场——123

原创

夏日のblog 于 2020-03-20 14:26:03 发布 904 收藏

分类专栏: [CTF-WEB](#)

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/zss192/article/details/104988778>

版权



[CTF-WEB 专栏收录该内容](#)

16 篇文章 0 订阅

订阅专栏

## 目录

[题目描述](#)

[writeup](#)

[小结](#)

## 题目描述

### “百度杯” CTF比赛 九月场

分值: 50分    类型: Web    题目名称: 123    已解答

题目内容: 12341234, 然后就解开了

本题来自播主C26

<http://6215f296ecd840719f21c3a77a5830515009709aa1a44ec7.changame.ichunqiu.com>

00 : 58 : 00

[延长时间\(3\)](#)    [重新创建](#)

<https://blog.csdn.net/zss192>

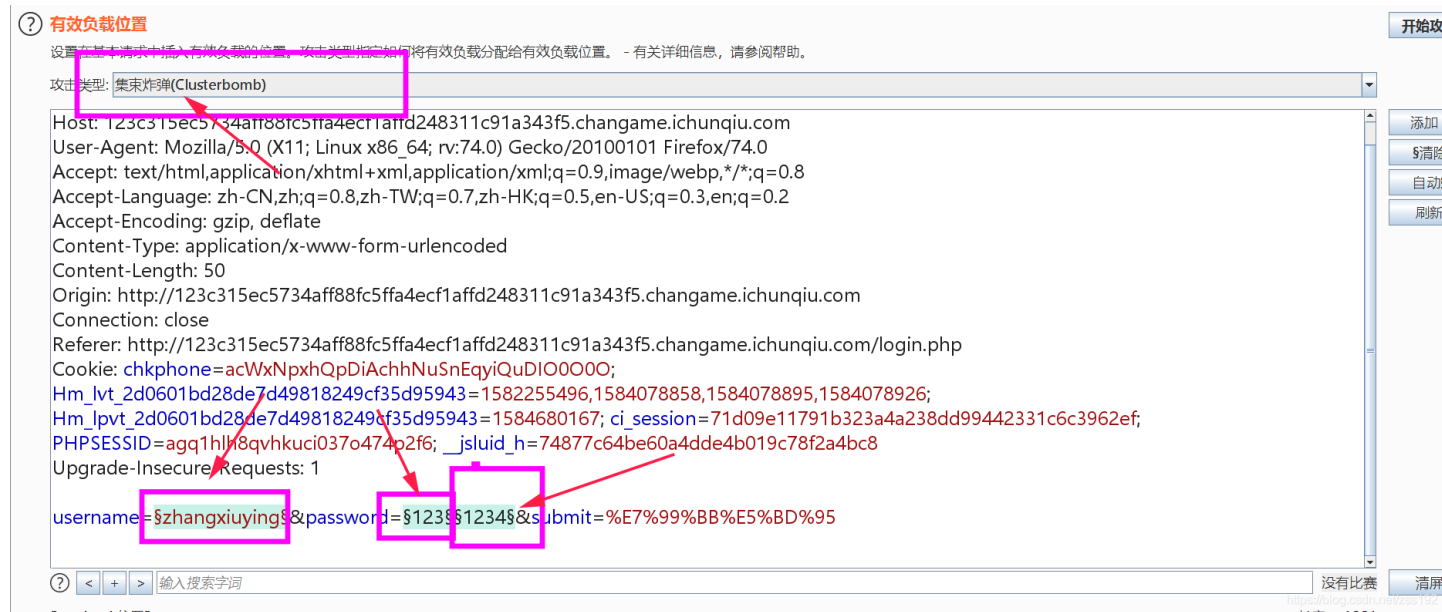
## writeup

打开页面是一个登录页面, 查看源码发现提示语句

```
<br /> <br />
<input type="submit" name="submit" value="登录" />

<!-- 用户信息都在user.php里 -->
<!-- 用户默认默认密码为用户名+出生日期 例如:zhangwei1999 -->
</form>
```

访问user.php发现界面为空，此时尝试打开备份文件，发现user.php.bak，下载后发现用户名，根据上述源码提示采用burpsuite爆破，配置如下，注意选取Clusterbomb模式（因为要尝试多个payload，详细见：Burp Suite Intruder4种攻击类型）



三个payload如下所示

**有效载荷集**  
您可以定义一个或多个有效载荷集。有效载荷集的数量取决于“位置”选项卡中定义的攻击类型。每个有效载荷

有效载荷集: 1      有效载荷数量: 357  
有效载荷类型: 简单清单      请求数量: 不明

**有效载荷选项[简单列表]**      选取刚才下载的用户.php.bak文件  
设置用于有效内容的简单字符串列表。

粘贴	zhangwei
载入中.....	wangwei
删除	wangfang
清屏	liwei
添加	lina
	zhangmin
	输入新项目

从列表中添加...

**有效载荷集**  
您可以定义一个或多个有效载荷集。有效载荷集的数量取决于“位置”选项卡中定义的攻击类型。每个有效载荷

有效载荷集: 2      有效载荷数量: 不明

有效载荷类型: 复制其他负载 请求数量: 不明

### ? 有效载荷选项[复制其他有效载荷]

此有效载荷类型将另一个有效负载的值复制到当前有效负载值。它可以与使用多个有效负载集的攻击类型一起使用。

来源位置: 1

### ? 有效负载处理

您可以定义在使用有效负载之前对每个有效负载执行各种处理任务的规则。

	效用	规则
添加		
编辑		
删除		

<https://blog.csdn.net/zss192>

您可以定义一个或多个有效负载集。有效负载集的数量取决于“位置”选项卡中定义的攻击类型。每个有效负载

有效负载集: 3 有效载荷数量: 41  
有效载荷类型: 数值 请求数量: 不明

### ? 有效载荷选项[数字]

生成给定范围内指定格式的数字有效内容。

#### 数字范围

类型:  连续  随机

From: 1980

To: 2020

增量: 1

编号:

自行尝试合适的范围

#### 数字格式

基地:  Decimal  Hex

整数部分的最小位数: 4

整数部分的最大位数: 4

少数民族最小位数: 0

少数最大数字: 0

#### 例

0001  
4321

<https://blog.csdn.net/zss192>

经过一番爆破后发现两个有用的信息

Results	Target	Positions	Payloads	Options
Filter: Showing all items				

Request	Payload1	Payload2	Payload3	Status	Error	Timeout	Length	Comment
3881	lixuiyun	lixuiyun	1990	200	<input type="checkbox"/>	<input type="checkbox"/>	1041	
5550	zhangyuzhen	zhangyuzhen	1995	200	<input type="checkbox"/>	<input type="checkbox"/>	1041	
0				200	<input type="checkbox"/>	<input type="checkbox"/>	1006	
2	wangwei	wangwei	1980	200	<input type="checkbox"/>	<input type="checkbox"/>	1006	
1	zhangwei	zhangwei	1980	200	<input type="checkbox"/>	<input type="checkbox"/>	1006	
4	liwei	liwei	1980	200	<input type="checkbox"/>	<input type="checkbox"/>	1006	
3	wangfang	wangfang	1980	200	<input type="checkbox"/>	<input type="checkbox"/>	1006	
6	zhangmin	zhangmin	1980	200	<input type="checkbox"/>	<input type="checkbox"/>	1006	
7	lijing	lijing	1980	200	<input type="checkbox"/>	<input type="checkbox"/>	1006	
8	wangjing	wangjing	1980	200	<input type="checkbox"/>	<input type="checkbox"/>	1006	
9	liuwei	liuwei	1980	200	<input type="checkbox"/>	<input type="checkbox"/>	1006	
10	wangxiuying	wangxiuying	1980	200	<input type="checkbox"/>	<input type="checkbox"/>	1006	

选取其中一个登录，发现依然是空白，查看源码发现

```

1 </body>
2 <center>
3 <!-- 存在漏洞需要去掉 -->
4 <!-- <form action="" method="POST" enctype="multipart/form-data">
5 <input type="file" name="file" />
6 <input type="submit" name="submit" value="上传" />
7 </form> -->
8 </center>
9 </body>
10 </html>

```

<https://blog.csdn.net/zss192>

我们可以看到提示这个表单存在漏洞，所以我们可以按F12然后把表单取消注释（选中<center>右键选择编辑html）

尝试上传一个正常jpg图片提示文件名不合法，猜想这里并不是真的文件上传，并不是用菜刀连上找flag。只是构造文件名，并且上传到服务器成为可执行文件便可通过。

Apache 配置文件中会有

```

+.ph(p[345]?|t|tml)
+.\.phps$

```

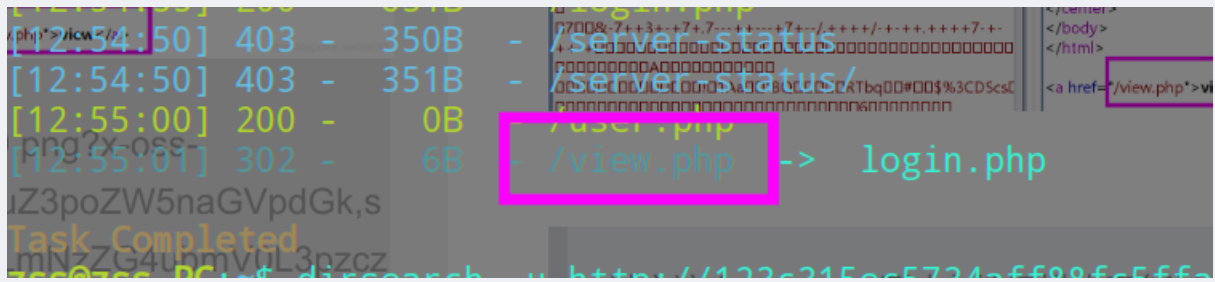
文件名满足即可被当做php解析，也就是说php3, php4, php5, pht, phtml,phps都是可以被解析的。

尝试.pht发现返回view.php

The image shows a browser's developer tools. On the left, the network tab displays a request body for a multipart form-data. The 'file' field has a filename of '1.jpg.pht' and a content-type of 'image/jpeg'. On the right, the response HTML is shown, featuring a title '个人中心' and a link to '/view.php' with the text 'view'.

<https://blog.csdn.net/zss192>

其实这个view.php通过文件扫描也可能被找到，我用的dirsearch就发现这个文件



进入view.php,提示file?,应该是file传参构造?file=flag.php

## filter "flag"

提示flag被过滤，尝试双写大小写也不行，最终发现构造

```
?file=flaflagg
```

发现flag

```
<?php
echo 'flag is here';
'flag{2364b0f7-824e-4741-9e89-d6b7964011b1}-';
?>
```

吐槽一句这个file也太迷惑人了，我以为必须是个文件才行。。。

## 小结

这道题涉及到了文件泄露、爆破、文件上传绕过、关键字绕过。非常好的一道题目，在此特别记录，文中如有错误，请联系我更正。