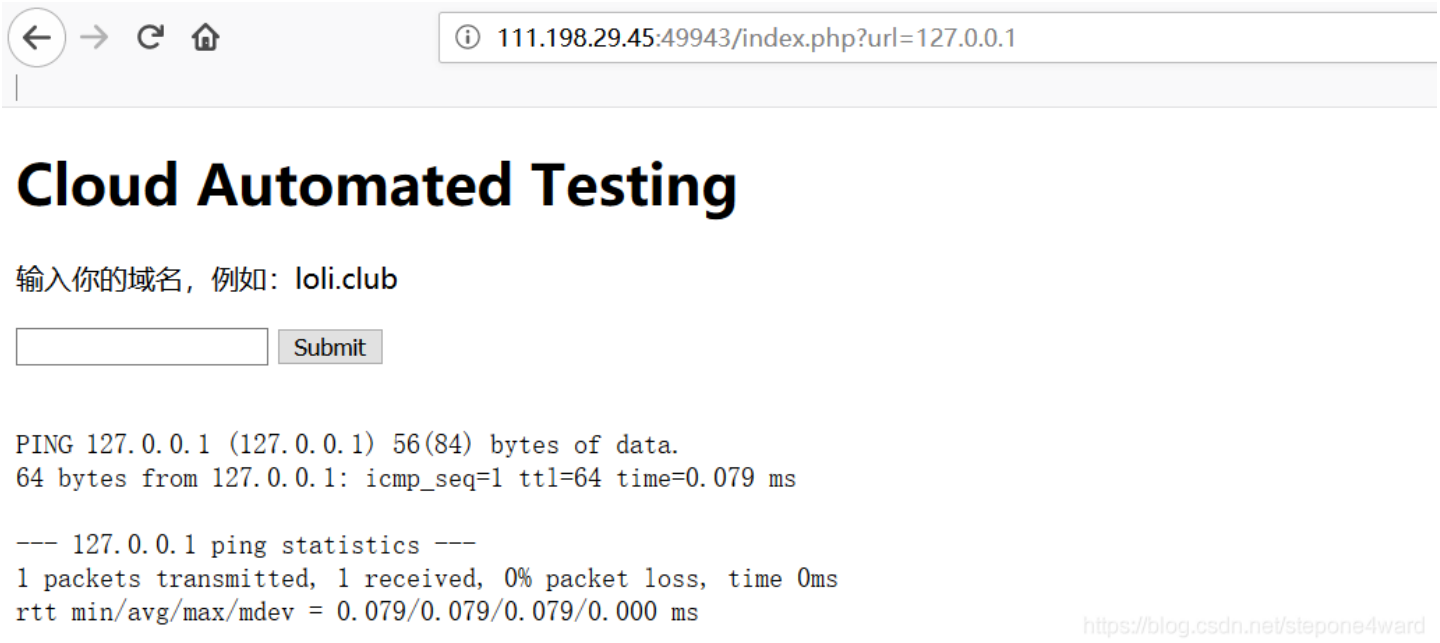
XCTF Cat writeup

原创

GAPPPPP 于 2019-07-04 14:53:25 发布 2948 收藏 3

刚看到题目的时候猜测是否是需要利用任意命令执行cat来读取文件的内容，后来才发现原来CAT是Cloud Automated Testing的缩写…



猜测1

测试输入127.0.0.1发现网站起到了检测目标站点网络是否连通(ping)的作用，我们都知道ping是一个php当中很危险的函数，可以利用 `|` 实现任意命令执行，测试payload: `127.0.0.1|phpinfo();` ,得到返回为 `Invalid URL` 猜测我们的输入当中存在有非法字符导致命令无法执行,fuzz一下可使用的字符只剩下了数字，英文字母和 `.` ，这么一想构造任意命令执行似乎无法实现了。

猜测2

之前做过一道MOCTF的名叫网站检测器的题目，利用到的是url解析差异实现的ssrf，那道题目当中实现了利用十六进制编码和url二次编码对于dot的绕过，但在本题当中并不适用，我们并不清楚想要读取到的网页名称，并且测试过后ssrf也并不存在。

正解

查看官方的wp开始的时候并不是很理解为什么输入为%80(以及%80之后的url编码)就可以返回Django报错，查看url编码表后

€	%80
	%81
‚	%82
ƒ	%83
„	%84
…	%85

†	%86
‡	%87
ˆ	%88
‰	%89
Š	%8a
‹	%8b
Œ	%8c
	%8d
Ž	%8e
	%8f

可以看到%80后的字符结合报错信息`UnicodeEncodeError`可以推断是由于ascii编码不支持导致的报错,根据报错信息可以得到的信息

```
Request Method: POST
Request URL: http://127.0.0.1:8000/api/ping

Django Version: 1.10.4
Python Version: 2.7.12
```

网站是使用Django进行开发的,结合PHP可以通过在参数中注入@来读取文件的漏洞,依次查看python的配置文件和数据库得到flag的内容

payload:`?url=@/opt/api/api/settings.py`,获取数据库名

```
os.path.join(BASE_DIR, \\\&#39;database.sqlite3\\\&#39;),
```

payload:`?url=@/opt/api/database.sqlite3`,获取数据库内容

```
c\x01\x02AWHCTF[redacted]}\n&#39;</pre></td
```