攻防世界-web-shrine-从0到1的解题历程writeup

原创

CTF小白 于 2020-04-13 13:49:36 发布 4316 收藏

分类专栏： CTF

本文链接：https://blog.csdn.net/qq_41429081/article/details/105487173

题目环境分析

首先开启靶机获取到题目如下

```
import flask
import os
app = flask.Flask(__name__)
app.config['FLAG'] = os.environ.pop('FLAG')

@app.route('/')
def index():
return open(__file__).read()

@app.route('/shrine/<path:shrine>')
def shrine(shrine):
  def safe_jinja(s):
    s = s.replace('(', '').replace(')', '')
    blacklist = ['config', 'self']
    return ''.join(['{{% set {}=None%}}'.format(c) for c in blacklist]) + s
  return flask.render_template_string(safe_jinja(shrine))

if __name__ == '__main__':
app.run(debug=True)
```

第一步对题目给出代码进行分析

```
@app.route('/')
def index():
return open(__file__).read()
```

默认访问路径为'/'，那么会将源代码读取出来，也就是默认页面所呈现的。

```
@app.route('/shrine/<path:shrine>')
def shrine(shrine):
  def safe_jinja(s):
    s = s.replace('(', '').replace(')', '')
    blacklist = ['config', 'self']
    return ''.join(['{{% set {}=None%}}'.format(c) for c in blacklist]) + s
  return flask.render_template_string(safe_jinja(shrine))
```

这边访问'/shrine/'路径下，传入一个值，返回的是

```
flask.render_template_string(safe_jinja(shrine))
```

我们输入的值首先被传到了safe_jinja函数，然后由flask.render_template_string进行渲染
对于没有用过flask框架的我来说理解safe_jinja函数最后的return操作有点困难，我本地执行了一下

很容易理解的是，我们传入的s会首先被去除'('，')',然后在最后加上处理后的s，前面是

{% set config=None%}{% set self=None%}

可以知道是结合flask.render_template_string渲染肯定会有漏洞。

解题过程

看到这边忽然才发现，flask最著名的不就是模板注入吗，直接模板注入不就行了。

发现果然可以执行。

但是我们之前已经发现他会将'(',')'替代。并且{% set config=None%}{% set self=None%}，将config和self加入了黑名单。

这边百度了一下flask模板注入绕过，发现并没有绕过括号的。。。。。

于是启动搜索writeup大法。

给出的payload为

```
{{get_flashed_messages.__globals__['current_app'].config['FLAG']}}
```