

HackTheBox-baby CachedView

原创

galaxy3000 于 2022-02-16 10:10:15 发布 968 收藏

分类专栏: [# Web](#) 文章标签: [CTF writeup](#) [网络安全](#) [hackthebox](#) [靶机](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/galaxy3000/article/details/122956671>

版权

Web

[Web](#) 专栏收录该内容

16 篇文章 0 订阅

订阅专栏

文章目录

概述

题目

题目概述

源代码

源代码分析

解题思路

题目解答

概述

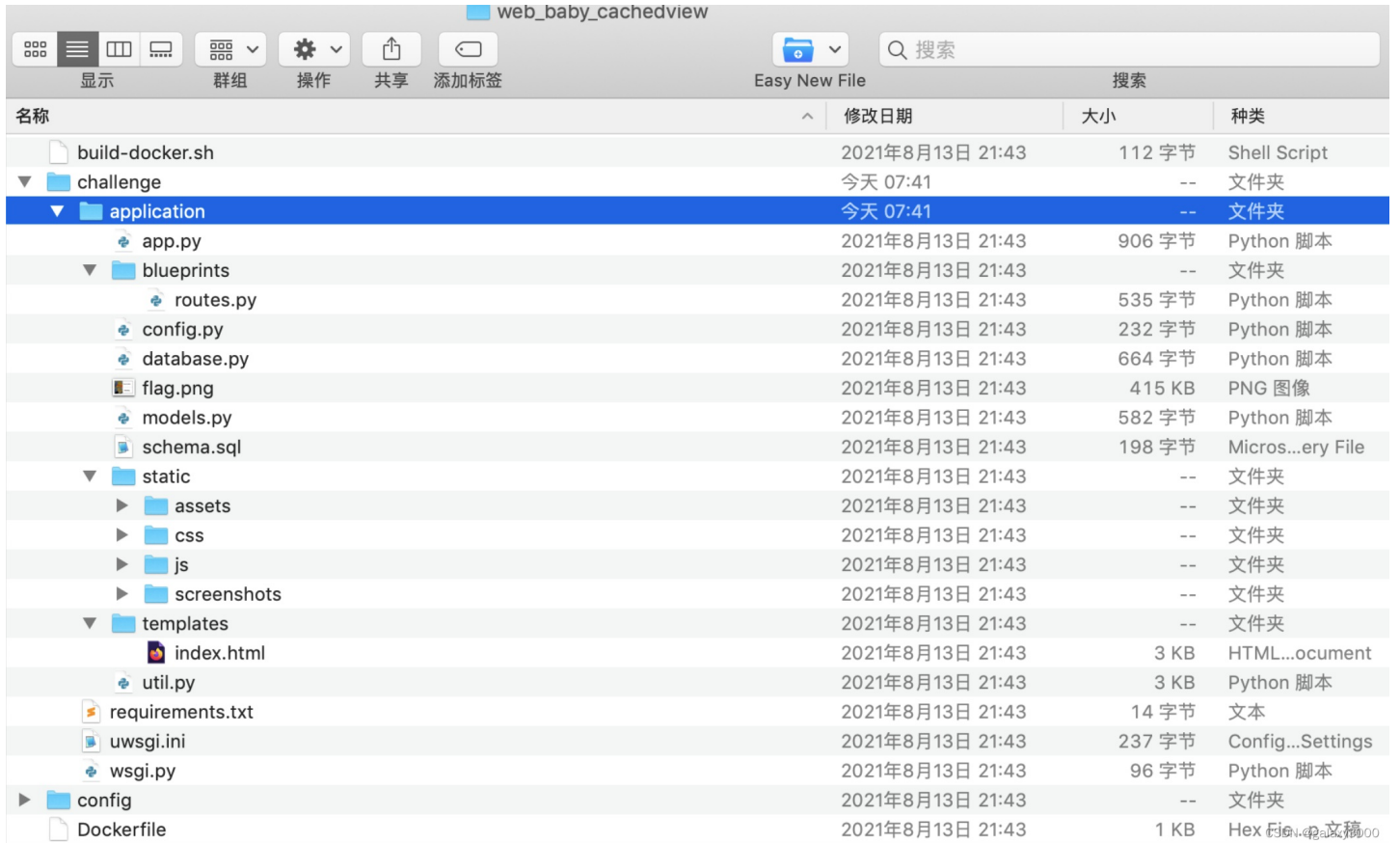
HackTheBox 网站CTF靶场Web相关题目baby CachedView, 题目地址<https://app.hackthebox.com/challenges/baby-cachedview>, 主要考察的是IP地址限制的绕过思路。

The screenshot shows the challenge page for 'baby CachedView' on HackTheBox. The challenge is categorized as 'EASY' and is worth 20 points. It is currently 'ONLINE'. The challenge description reads: 'I made a service for people to cache their favourite websites, come and check it out! But don't try anything funny, after a recent incident we implemented military grade IP based restrictions to keep the hackers at bay...'. The host IP is listed as 167.99.202.131:30522. The challenge has a rating of 188 likes and 11 dislikes, and has been solved by 749 users. The category is 'Web'. There are buttons for 'Stop Instance', 'Download Files', and 'Share Results'. The page also shows a 'ZIP PASSWORD' field and a 'CSDN @galaxy3000' watermark.

题目

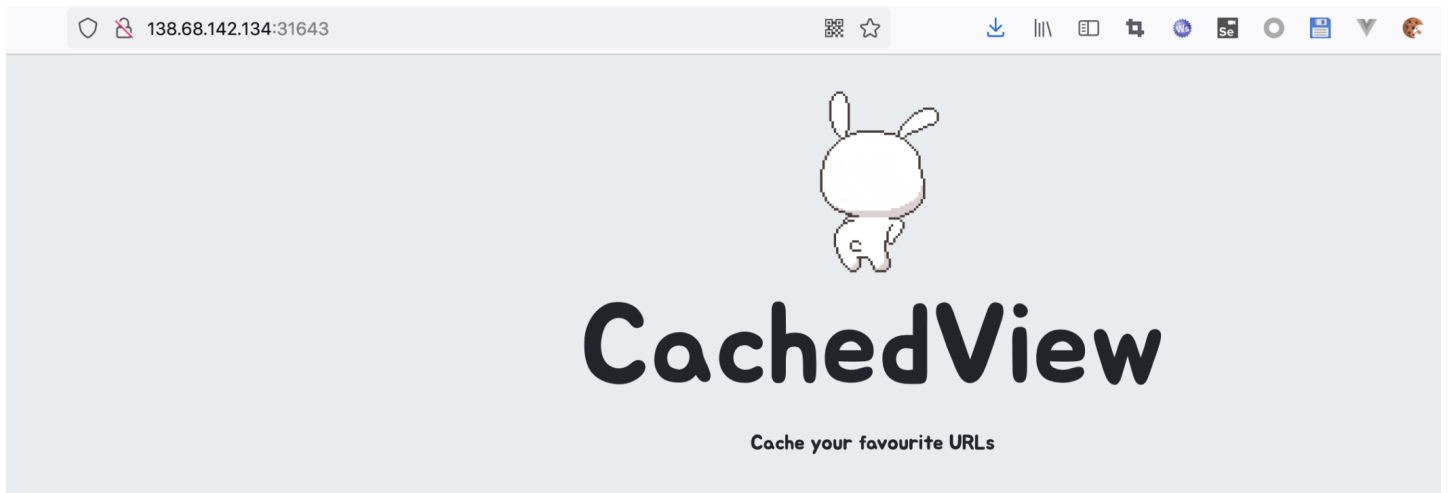
题目概述

题目提供了附件下载，解压后如图



题目提示大意为经历了安全事故后对网站的服务进行了基于IP地址的限制。

开启程序实例后，提示访问 `138.68.142.134:31643`，访问 `http://138.68.142.134:31643`，看到如下Web界面



CSDN @galaxy3000

查看requirements.txt，发现为Flask应用

Flask
selenium

源代码

routes.py

```
from flask import Blueprint, request, render_template, abort, send_file
from application.util import cache_web, is_from_localhost

web = Blueprint('web', __name__)
api = Blueprint('api', __name__)

@web.route('/')
def index():
    return render_template('index.html')

@api.route('/cache', methods=['POST'])
def cache():
    if not request.is_json or 'url' not in request.json:
        return abort(400)

    return cache_web(request.json['url'])

@web.route('/flag')
@is_from_localhost
def flag():
    return send_file('flag.png')
```

可以在路由中看到 `/flag` 路径，这个路由有两个装饰器，其中用到了 `util.py`

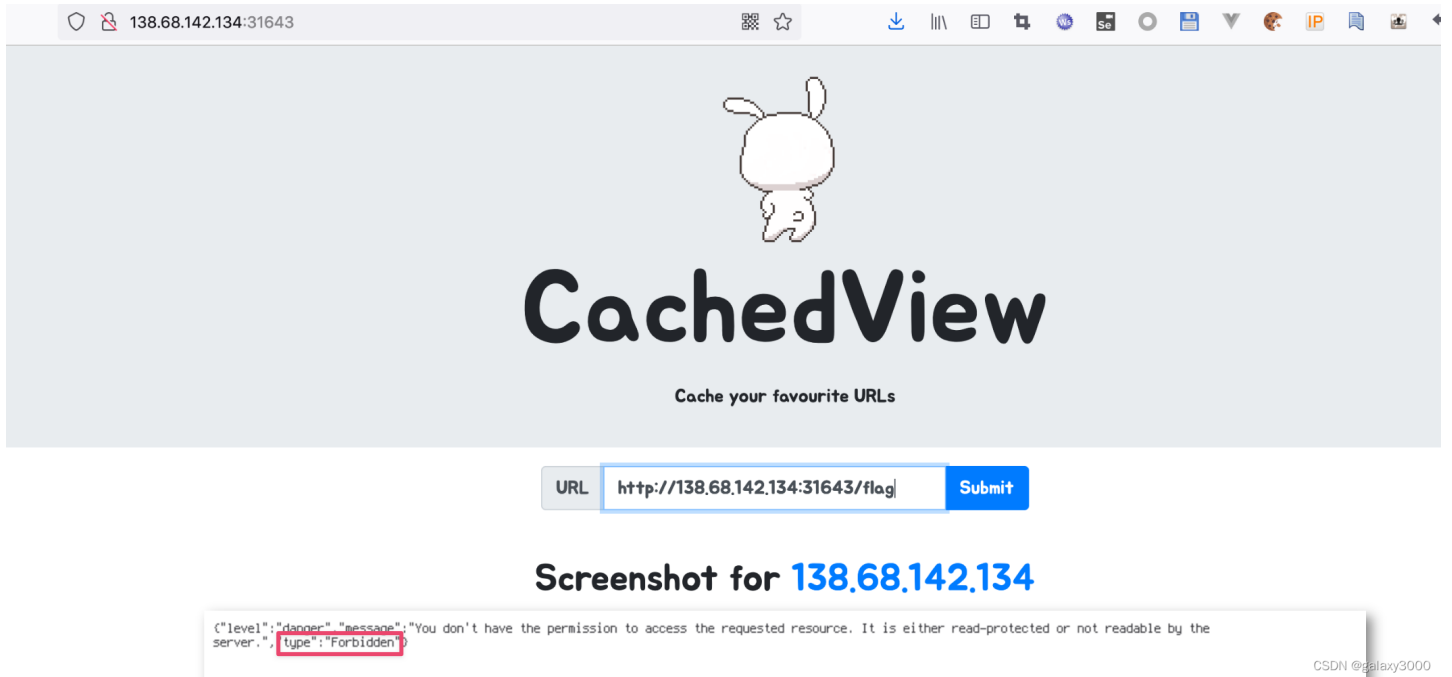
```
1 def cache_web(url):
2     scheme = urlparse(url).scheme
3     domain = urlparse(url).hostname
4
5     if not domain or not scheme:
6         return flash(f'Malformed url {url}', 'danger')
7
8     if scheme not in ['http', 'https']:
9         return flash('Invalid scheme', 'danger')
10
11     def ip2long(ip_addr):
12         return struct.unpack('!L', socket.inet_aton(ip_addr))[0]
13
14     def is_inner_ipaddress(ip):
15         ip = ip2long(ip)
16         return ip2long('127.0.0.0') >> 24 == ip >> 24 or \
17             ip2long('10.0.0.0') >> 24 == ip >> 24 or \
18             ip2long('172.16.0.0') >> 20 == ip >> 20 or \
19             ip2long('192.168.0.0') >> 16 == ip >> 16 or \
20             ip2long('0.0.0.0') >> 24 == ip >> 24
21
22     if is_inner_ipaddress(socket.gethostbyname(domain)):
23         return flash('IP not allowed', 'danger')
24
25     return serve_screenshot_from(url, domain)
26
27 def is_from_localhost(func):
28     @functools.wraps(func)
29     def check_ip(*args, **kwargs):
30         if request.remote_addr != '127.0.0.1' or request.referrer:
31             return abort(403)
32         return func(*args, **kwargs)
33     return check_ip
```

源代码分析

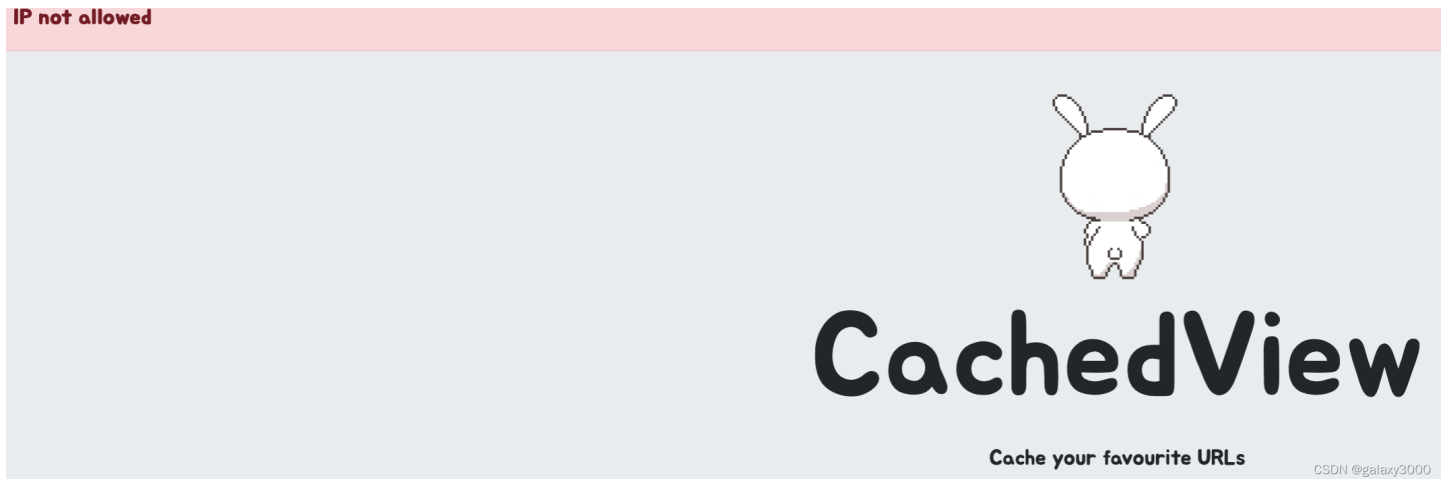
在目标首页提交的源代码，由 `/cache` 响应，经过 `cache_web` 函数处理后返回结果，在 `cache_web` 函数中有IP地址的限制，主要是对127网段、172.16网段、192.168等内网网段的限制。

而如果访问 `/flag` 路径，会被装饰器中的 `check_ip` 处理，如果不是本地地址（127.0.0.1）会被禁止。

实际验证，首先在输入框输入http://138.68.142.134:31643/flag，触发了 `check_ip` 的检查



如果在输入框输入http://127.0.0.1:31643/flag，触发了 `cache_web` 的检查，提示 `IP not allowed`



解题思路

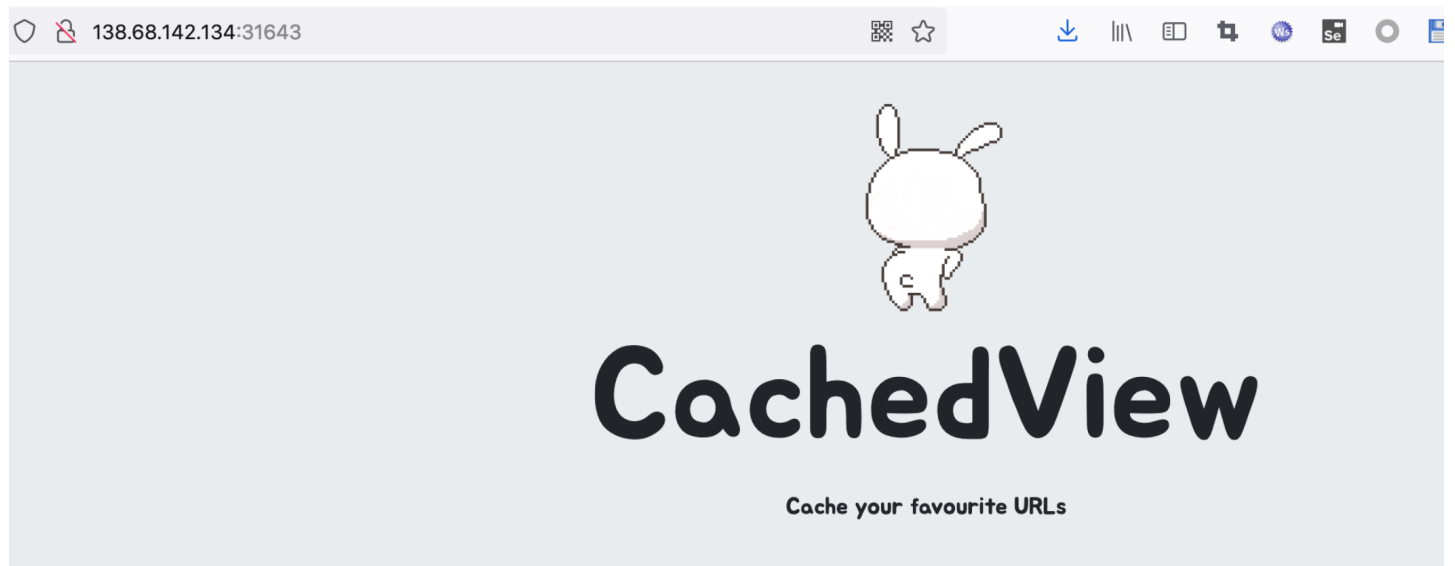
在云服务器上部署Web服务，输入框输入地址为云服务器地址，通过 `cache_web` 的检查，而服务器上Web的内容为本地访问flag

题目解答

在云服务器上部署网页index.html，内容为

```
<meta http-equiv="refresh" content="0; URL=http://127.0.0.1:31643/flag" />
```

在输入框输入云服务器地址



CSDN @galaxy3000

得到flag

Screenshot for 12 12

