🪟

baby Cached View

Platform	HTB
Date	@July 29, 2022
Operating System	Web-CTF
Tags	dns-rebinding python toctou web-app

General-Information

▼ Table of Contents

▼ Challenge Description

- I made a service for people to cache their favorite 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…

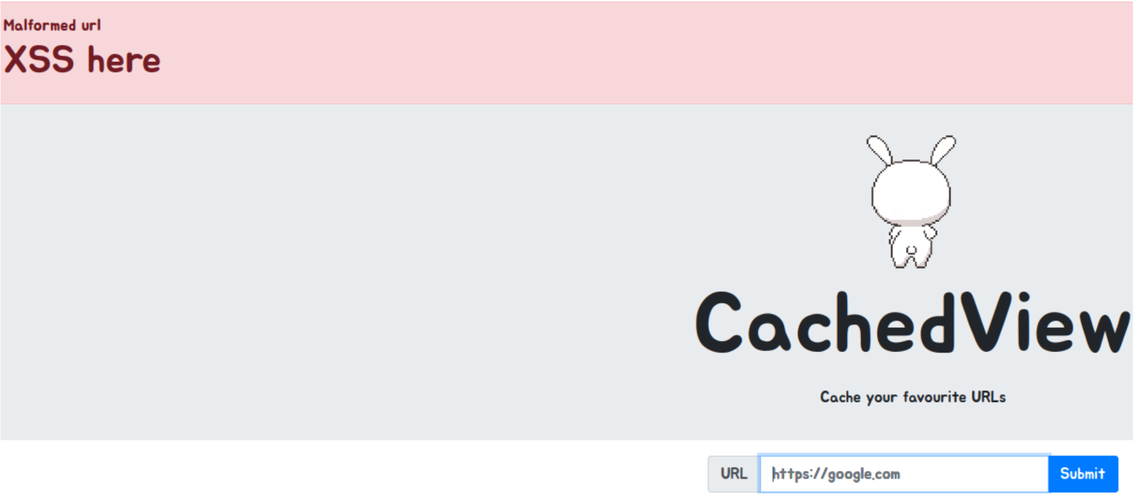
Summary

- A website that's vulnerable to a `TOCTOU` vulnerability that when leveraged with a DNS rebind allows you to view the flag for the challenge

---

Website

- ▼ When using the website I noticed that it returned whatever unsanitized input I gave the input field, which of course left it vulnerable to an XSS. I didn’t try to much after that because I wanted to go read the files and see where my input was going and what it was touching.
    - ▼ `<h1>XSS here</h1>`

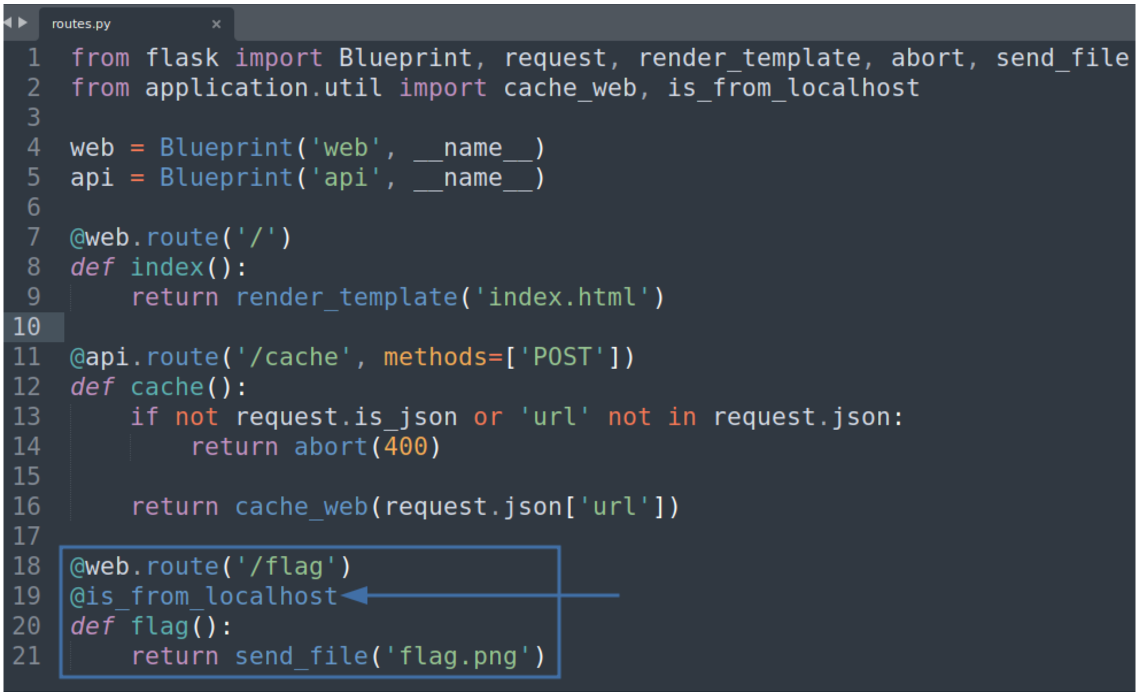


Malformed url

XSS here

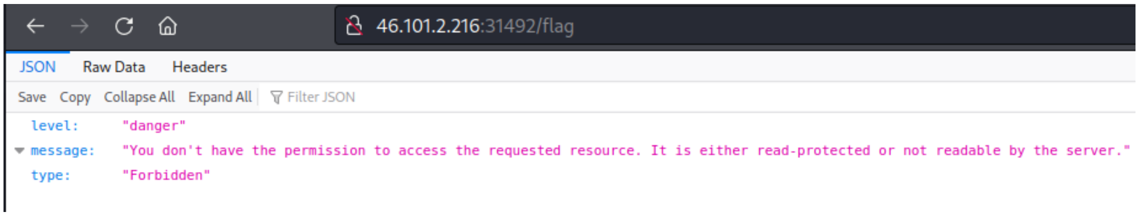
Vulnerability Identification

- ▼ At first after reading through the files a bit and noticing that in `routes.py` there’s another `web.route` called `/flag` which could be interesting. So I first tried to navigate to it and see what would happen.
    - ▼ `routes.py`



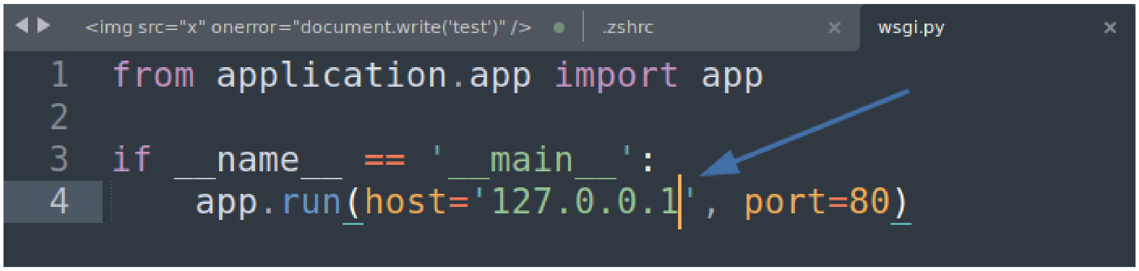
```python
from flask import Blueprint, request, render_template, abort, send_file
from application.util import cache_web, is_from_localhost

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')
```

- However, all I got was a 403 error message because as mentioned in the code, the server must `@is_from_localhost` or running on your localhost in order to appear. I first tried running the `build-docker.sh` file, which didn't work.
    - 403 Error Message



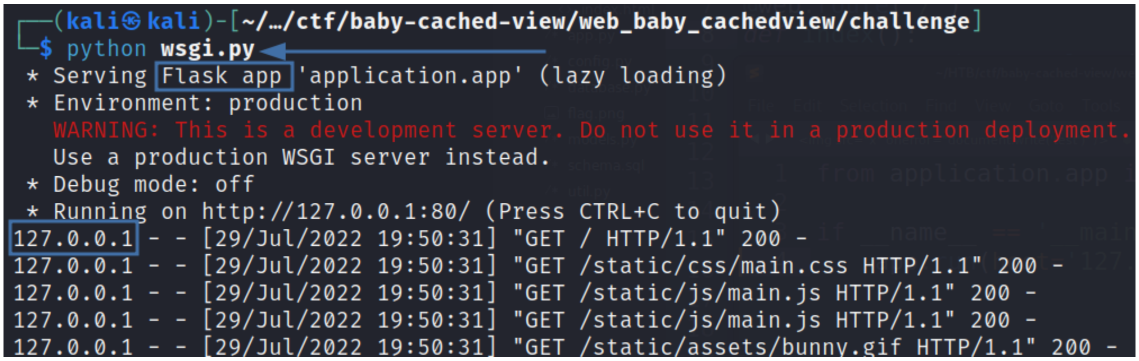
```
level:      "danger"
message:    "You don't have the permission to access the requested resource. It is either read-protected or not readable by the server."
type:       "Forbidden"
```

- So I turned to running the `wsgi.py` file because WSGI is used for Python web servers and this is a Flask application. Which worked and I was able to see the flag picture that lead me to look into `TOCTOU`'s or `Time of check to Time of use` attacks.
    - Changing the `wsgi.py` file to run from localhost and not 0.0.0.0



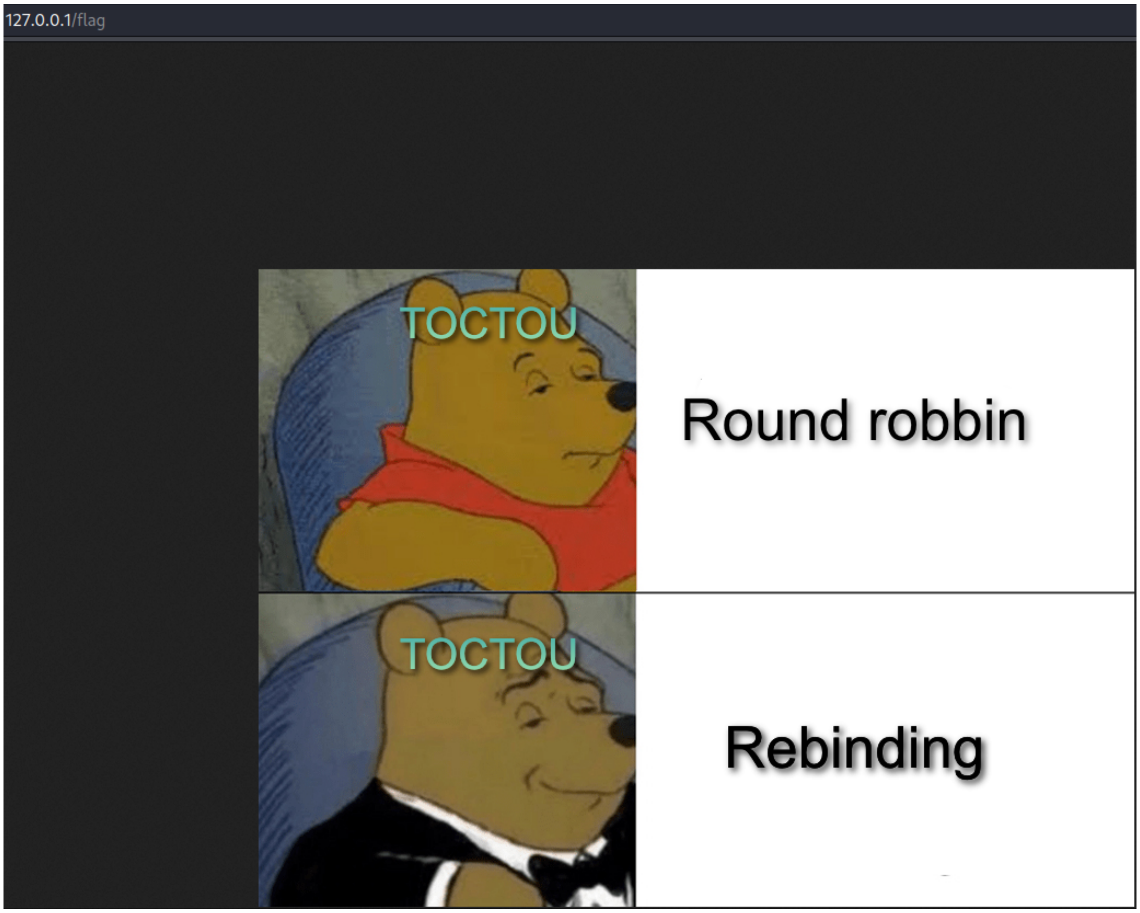
```python
from application.app import app

if __name__ == '__main__':
    app.run(host='127.0.0.1', port=80)
```

▼ Running `wsgi.py`



```
┌──(kali㉿kali)-[~/…/ctf/baby-cached-view/web_baby_cachedview/challenge]
└─$ python wsgi.py
 * Serving Flask app 'application.app' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:80/ (Press CTRL+C to quit)
127.0.0.1 - - [29/Jul/2022 19:50:31] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [29/Jul/2022 19:50:31] "GET /static/css/main.css HTTP/1.1" 200 -
127.0.0.1 - - [29/Jul/2022 19:50:31] "GET /static/js/main.js HTTP/1.1" 200 -
127.0.0.1 - - [29/Jul/2022 19:50:31] "GET /static/js/main.js HTTP/1.1" 200 -
127.0.0.1 - - [29/Jul/2022 19:50:31] "GET /static/assets/bunny.gif HTTP/1.1" 200 -
```

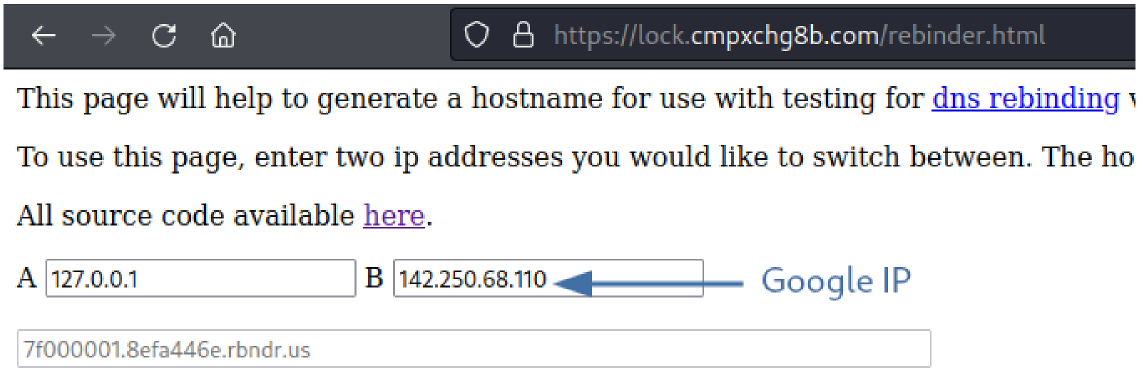
▼ Flag Image



DNS Rebinding

▼ I decided to do some googling into the `rebinding` that was being talked about on `flag.png` when I came across this DNS rebinding program. Which was going be to used to exploit the website since it would allow to requests to be sent to the website. The first request would be normal, but the second one would be the actual request querying for the flag!

▼ DNS Rebinding Program *(Used google because there was no need for it to run locally and it wouldn't run good locally lol)*



https://lock.cmpxchg8b.com/rebinder.html

This page will help to generate a hostname for use with testing for dns rebinding v

To use this page, enter two ip addresses you would like to switch between. The ho

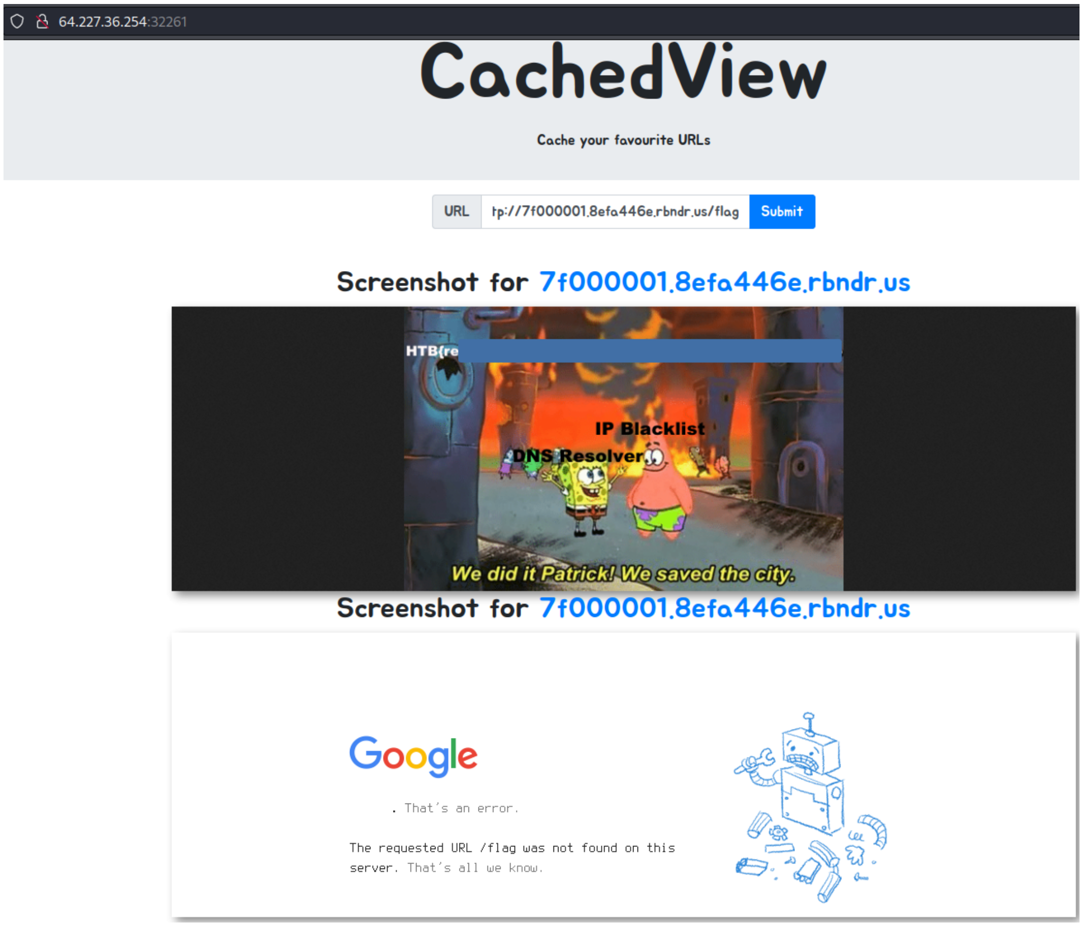
All source code available here.

A 127.0.0.1 B 142.250.68.110

7f000001.8efa446e.rbndr.us

▼ Flag

- `http://http://7f000001.8efa446e.rbndr.us/flag`



64.227.36.254:32261

CachedView
Cache your favourite URLs
URL
Submit

tp://7f000001.8efa446e.rbndr.us/flag

Screenshot for 7f000001.8efa446e.rbndr.us

HTB{re [redacted]

IP Blacklist
DNS Resolver
We did it Patrick! We saved the city.

Screenshot for 7f000001.8efa446e.rbndr.us

. That's an error.

The requested URL /flag was not found on this server. That's all we know.

- Also, keep in mind that sometimes you might just have to keep entering that same dns-rebinded string until the errors in `utils.py` are passed.

Information Learned

- I didn’t know anything about `TOCTOU`'s or `dnsrebinding` before this challenge, so getting a high-level iceberg taste of it was fun. Although it was hard at first to identify where the DNS rebinding string should be because of the errors at first.
- When working on a challenge and you think you know what the exploit can be, look up `$exploit-name ctf` and then you can find writeups to read through. Which can help you transfer that thinking into the challenge you’re working on.