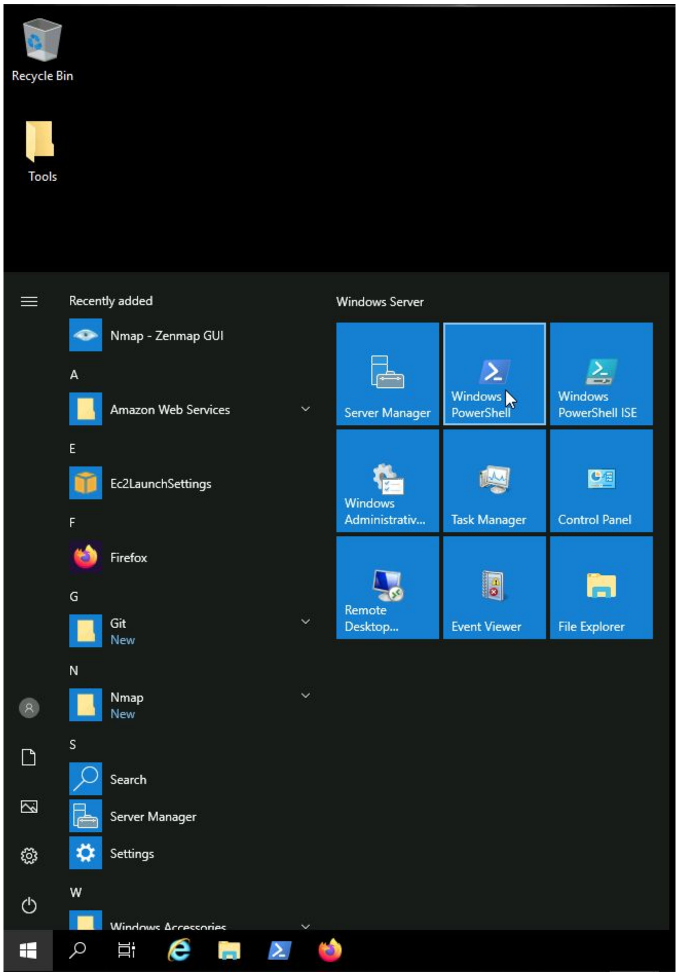
ATTACK DEFENSE

by PentesterAcademy

Name	Windows Recon: PowerShell Network Scanner
URL	https://attackdefense.com/challengedetails?cid=2217
Type	Windows Reconnaissance: Host Discovery

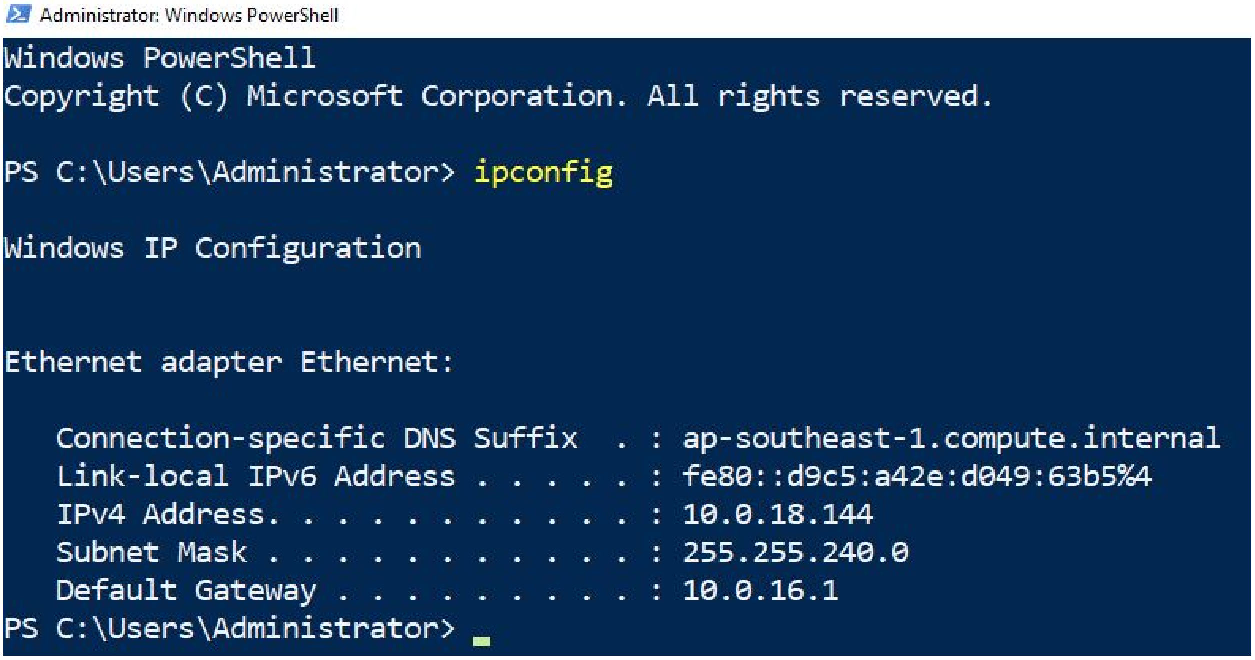
Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Step 1: Open powershell.exe terminal.



Step 2: Checking the IP address.

Command: ipconfig



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . : ap-southeast-1.compute.internal
   Link-local IPv6 Address . . . . . : fe80::d9c5:a42e:d049:63b5%4
   IPv4 Address. . . . . . . . . . . : 10.0.18.144
   Subnet Mask . . . . . . . . . . . : 255.255.240.0
   Default Gateway . . . . . . . . . : 10.0.16.1
PS C:\Users\Administrator> _
```

In this lab, we are going to run a PowerShell script for network scanning and host discovery. This kind of script or a manual technique of host, port discovery using PowerShell or windows command prompt is useful where we don’t have access to tools such as Nmap, Zenmap, Masscan, etc.

PowerShell | IPv4 network scanner

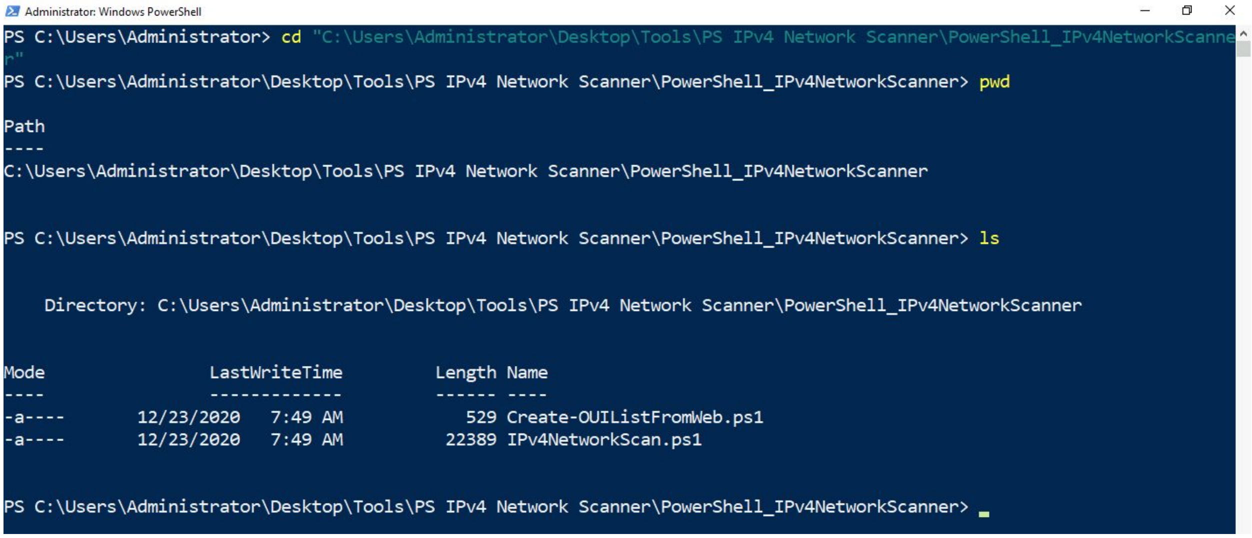
“This powerful asynchronous IPv4 network scanner for PowerShell allows you to scan every IPv4 range you want. But there is also the possibility to scan an entire subnet based on an IPv4 address within the subnet and a subnet mask/CIDR.”

Source: https://github.com/BornToBeRoot/PowerShell_IPv4NetworkScanner

Step 2: The scripts are located in “**C:\Users\Administrator\Desktop\Tools\PS IPv4 Network Scanner\PowerShell_IPv4NetworkScanner**” folder

Switch to the IPv4 Network Scanner folder.

Commands: cd "C:\Users\Administrator\Desktop\Tools\PS IPv4 Network Scanner\PowerShell_IPv4NetworkScanner"
pwd
ls



Administrator: Windows PowerShell

```
PS C:\Users\Administrator> cd "C:\Users\Administrator\Desktop\Tools\PS IPv4 Network Scanner\PowerShell_IPv4NetworkScanne
r"
PS C:\Users\Administrator\Desktop\Tools\PS IPv4 Network Scanner\PowerShell_IPv4NetworkScanner> pwd

Path
----
C:\Users\Administrator\Desktop\Tools\PS IPv4 Network Scanner\PowerShell_IPv4NetworkScanner


PS C:\Users\Administrator\Desktop\Tools\PS IPv4 Network Scanner\PowerShell_IPv4NetworkScanner> ls


    Directory: C:\Users\Administrator\Desktop\Tools\PS IPv4 Network Scanner\PowerShell_IPv4NetworkScanner


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        12/23/2020   7:49 AM            529 Create-OUIListFromWeb.ps1
-a----        12/23/2020   7:49 AM          22389 IPv4NetworkScan.ps1


PS C:\Users\Administrator\Desktop\Tools\PS IPv4 Network Scanner\PowerShell_IPv4NetworkScanner> _
```

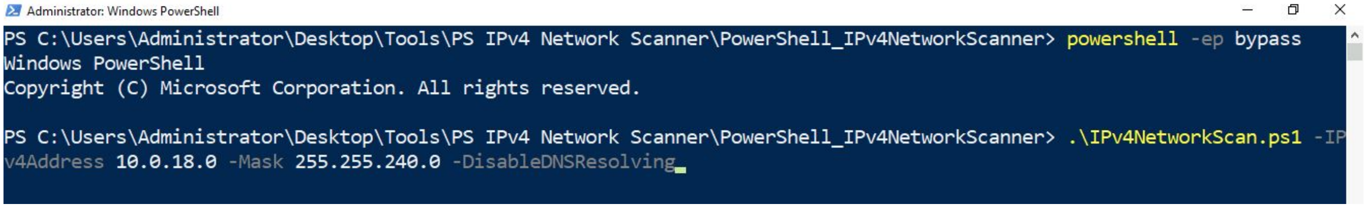
We can notice that there are two scripts. We will be using the “IPv4NetworkScan.ps1” PowerShell script for network subnet scanning.

Step 3: Scanning the entire range and disable DNS resolve by feeding the option: “DisableDNSResolving”

Commands: powershell -ep bypass

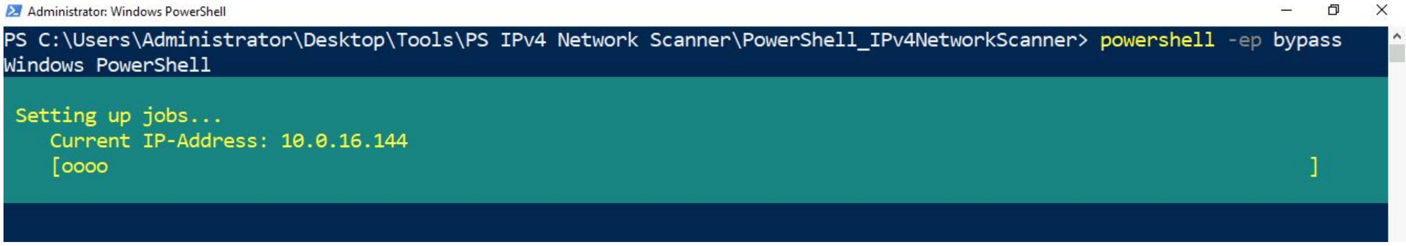
We are running “powershell -ep bypass” command to allow the current PS terminal to execute PowerShell script.

.\IPv4NetworkScan.ps1 -IPv4Address **10.0.18.0** -Mask 255.255.240.0 -DisableDNSResolving

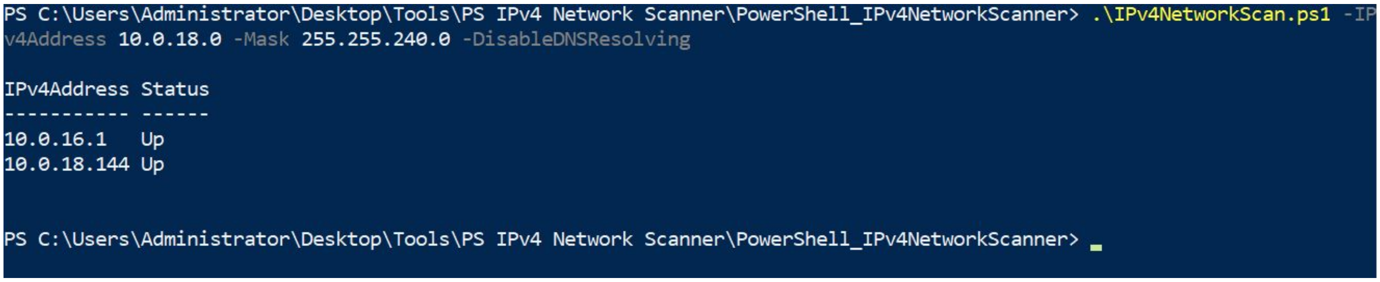


```
PS C:\Users\Administrator\Desktop\Tools\PS IPv4 Network Scanner\PowerShell_IPv4NetworkScanner> powershell -ep bypass
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator\Desktop\Tools\PS IPv4 Network Scanner\PowerShell_IPv4NetworkScanner> .\IPv4NetworkScan.ps1 -IP
v4Address 10.0.18.0 -Mask 255.255.240.0 -DisableDNSResolving
```

Note: The scanning would take 3-5 minutes.



```
PS C:\Users\Administrator\Desktop\Tools\PS IPv4 Network Scanner\PowerShell_IPv4NetworkScanner> powershell -ep bypass
Windows PowerShell

 Setting up jobs...
    Current IP-Address: 10.0.16.144
    [oooo                                                                                  ]
```

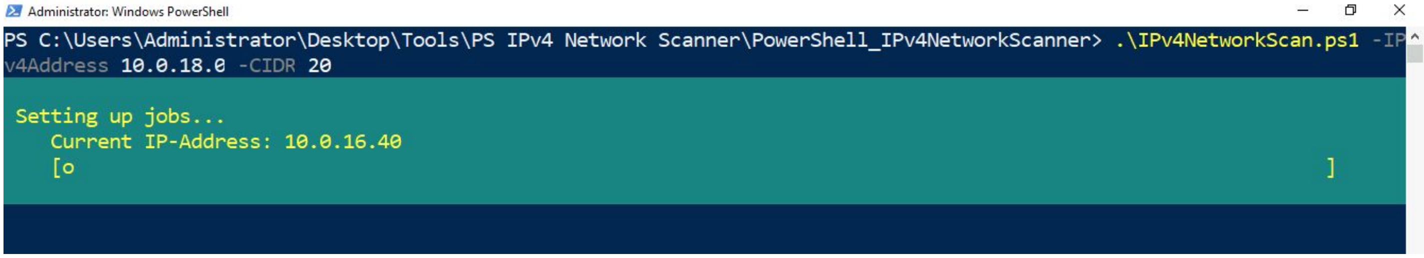


```
PS C:\Users\Administrator\Desktop\Tools\PS IPv4 Network Scanner\PowerShell_IPv4NetworkScanner> .\IPv4NetworkScan.ps1 -IP
v4Address 10.0.18.0 -Mask 255.255.240.0 -DisableDNSResolving

IPv4Address Status
----------- ------
10.0.16.1   Up
10.0.18.144 Up


PS C:\Users\Administrator\Desktop\Tools\PS IPv4 Network Scanner\PowerShell_IPv4NetworkScanner>
```

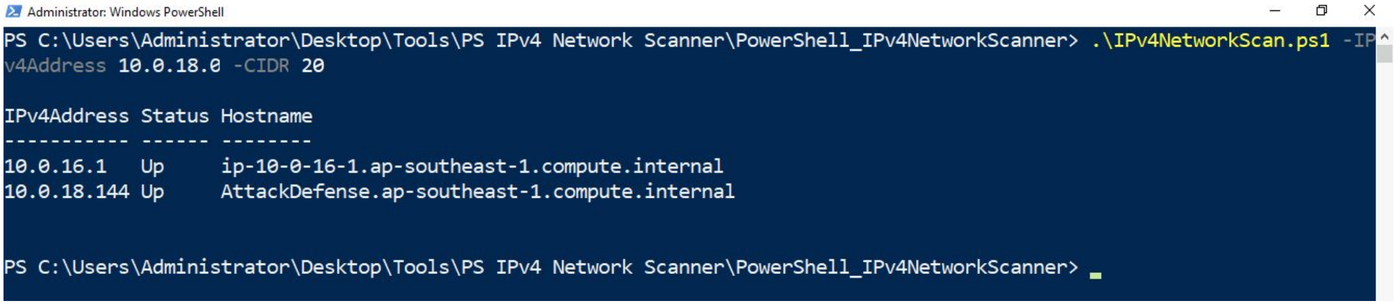
We can observe, we have discovered two hosts using the PS Script.

Step 4: Rescan the network again and discover the hostname of the alive machines.

Command: .\IPv4NetworkScan.ps1 -IPv4Address 10.0.18.0 -CIDR 20



```
PS C:\Users\Administrator\Desktop\Tools\PS IPv4 Network Scanner\PowerShell_IPv4NetworkScanner> .\IPv4NetworkScan.ps1 -IP
v4Address 10.0.18.0 -CIDR 20

 Setting up jobs...
    Current IP-Address: 10.0.16.40
    [o                                                                                     ]
```



```
PS C:\Users\Administrator\Desktop\Tools\PS IPv4 Network Scanner\PowerShell_IPv4NetworkScanner> .\IPv4NetworkScan.ps1 -IP
v4Address 10.0.18.0 -CIDR 20

IPv4Address Status Hostname
----------- ------ --------
10.0.16.1   Up     ip-10-0-16-1.ap-southeast-1.compute.internal
10.0.18.144 Up     AttackDefense.ap-southeast-1.compute.internal


PS C:\Users\Administrator\Desktop\Tools\PS IPv4 Network Scanner\PowerShell_IPv4NetworkScanner>
```

We have discovered the target machine’s hostname.

References:

1. PS Network Scanner (https://github.com/BornToBeRoot/PowerShell_IPv4NetworkScanner)