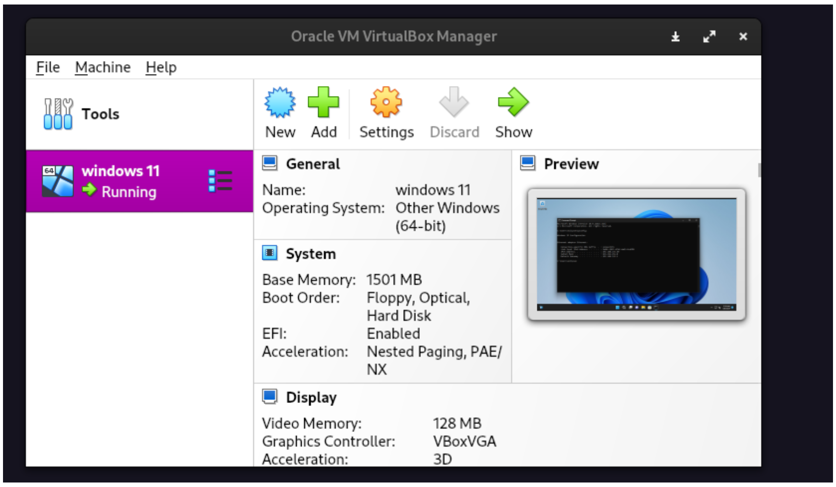
# HOST DISCOVERY

Peter Maina

September 6, 2024

# 1 Host Discovery

We have created a virtual box of a Windows 11 OS (VICTIM) and bridged our Ethernet connection to the box.



Name: windows 11
Operating System: Other Windows (64-bit)

Base Memory: 1501 MB
Boot Order: Floppy, Optical, Hard Disk
EFI: Enabled
Acceleration: Nested Paging, PAE/NX

Video Memory: 128 MB
Graphics Controller: VBoxVGA
Acceleration: 3D

Figure 1: Virtual Machine Setup

The host machine is a KALI LINUX machine (ATTACKER). We own both systems and we are using our network for any penetration testing.

From our KALI we will try to discover the victim machine's IP address. Since the victim is using the same network as the host, we will first discover our own IP.

## 1.1 Discovering Our Own IP Address

```
ifconfig
```

```
ap0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.12.1  netmask 255.255.255.0  broadcast 192.168.12.255
        inet6 fe80::6e88:14ff:fe18:8861  prefixlen 64  scopeid 0x20<link>
        ether 6c:88:14:18:88:61  txqueuelen 1000  (Ethernet)
        RX packets 144554  bytes 22666321 (21.6 MiB)
        RX errors 0  dropped 3  overruns 0  frame 0
        TX packets 148332  bytes 170053826 (162.1 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet \textcolor{red}{192.168.215.102}  netmask 255.255.255.0  broadcast
        inet6 fe80::470c:4a4c:6053:57b8  prefixlen 64  scopeid 0x20<link>
        ether 2c:59:e5:ba:4e:62  txqueuelen 1000  (Ethernet)
        RX packets 512022  bytes 563671282 (537.5 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 539695  bytes 66889016 (63.7 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 17  memory 0xd4400000-d4420000

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 38952  bytes 3984120 (3.7 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 38952  bytes 3984120 (3.7 MiB)
        TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

wlan0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 6c:88:14:18:88:60  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

We will need the Ethernet (`eth0`) IP.

We can use a tool called `ipcalc` to see the IP subnet mask range of 192.168.215.......

```
ipcalc 192.168.215.102
```

```
Address:   192.168.215.102      11000000.10101000.11010111. 01100110
```

```
Netmask:   255.255.255.0 = 24   11111111.11111111.11111111. 00000000
Wildcard:  0.0.0.255            00000000.00000000.00000000. 11111111
=>
Network:   192.168.215.0/24     11000000.10101000.11010111. 00000000
HostMin:   192.168.215.1        11000000.10101000.11010111. 00000001
HostMax:   192.168.215.254      11000000.10101000.11010111. 11111110
Broadcast: 192.168.215.255      11000000.10101000.11010111. 11111111
Hosts/Net: 254                   Class C, Private Internet
```

From the output, the subnet mask seems to be 24.

## 1.2 Using Nmap

```
sudo nmap -PE -sn 192.168.215.0/24 -oN hostdiscovery.txt

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-06 14:55 EDT
Nmap scan report for 192.168.215.1
Host is up (0.00081s latency).
MAC Address: A4:4C:C8:50:E9:1E (Dell)
Nmap scan report for 192.168.215.50
Host is up (0.00021s latency).
MAC Address: 08:00:27:42:0A:D3 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.215.102
Host is up.
Nmap done: 256 IP addresses (3 hosts up) scanned in 27.99 seconds
```

From the output, we have 3 hosts discovered, and our target is given as `192.168.215.50`. Since it is hosted on the VirtualBox, the rest are our network source and the local machine (KALI).

We have successfully identified our host using NMAP.

## 1.3 Using Bettercap

Bettercap is a network discovery and sniffing tool that can sniff on devices connected to our network, so it can do more than host discovery.

```
[sudo] password for eclipse:
bettercap v2.32.0 (built for linux amd64 with go1.22.3) [type 'help' for a list

192.168.215.0/24 > 192.168.215.102  » [15:05:03] [sys.log] [inf] gateway monitor
192.168.215.0/24 > 192.168.215.102  » net.probe on
```

```
192.168.215.0/24 > 192.168.215.102  » [15:05:10] [sys.log] [inf] net.probe start
192.168.215.0/24 > 192.168.215.102  » [15:05:10] [sys.log] [inf] net.probe probi
192.168.215.0/24 > 192.168.215.102  » [15:05:10] [endpoint.new] endpoint 192.168
192.168.215.0/24 > 192.168.215.102  » net.show


       IP                 MAC                 Name                   Vendor

  192.168.215.102  2c:59:e5:ba:4e:62  eth0                Hewlett Packard
  192.168.215.1    a4:4c:c8:50:e9:1e  gateway             Dell Inc.

  192.168.215.50   08:00:27:42:0a:d3  ADMIN.connectify.   PCS Computer Systems G


↑ 14 kB / ↓ 211 kB / 986 pkts

192.168.215.0/24 > 192.168.215.102  »
```

From Bettercap, we can see our hosts but we can't tell the target's IP. Since we have a fewer number of hosts, we can eliminate our local machine and/or network gateway. Bettercap also gave us the target's name as `ADMIN.connectify`, which can be used for further attacks.

## 2 Conclusion

From this, we have successfully performed host discovery using two tools.