Lab - Introduction to Nmap

Overview

Network Mapper (Nmap) is a network scanning and host detection tool that is very useful during several steps of penetration testing. Nmap is not limited to merely gathering information and enumeration, it is also a powerful utility that can be used as a vulnerability detector or a security scanner. Nmap is a multipurpose tool; it can be run on many different operating systems including Windows, Linux, BSD, and Mac. Nmap can be used to perform the following:

- Detect any live hosts present on the network (host discovery)
- Detect any open ports on a host (port discovery or enumeration)
- Detect software of the respective port assigned to the service (service discovery)
- Detect the operating system, hardware address, and the software version
- Detect any vulnerabilities and security holes using Nmap scripts

Nmap is a very common tool, and it is available for both the command line interface and the graphical user interface. The objective of this lab is to create a guide containing information about Nmap and its usage.

- Introduction to Nmap
- Learn Nmap parameters and techniques of scanning
- Introduction to operating system detection
- Nmap tutorial

Nmap uses different techniques to perform scanning including TCP connect () scanning, TCP reverse ident scanning, FTP bounce scanning, and so on. All these types of scanning have their advantages and disadvantages.

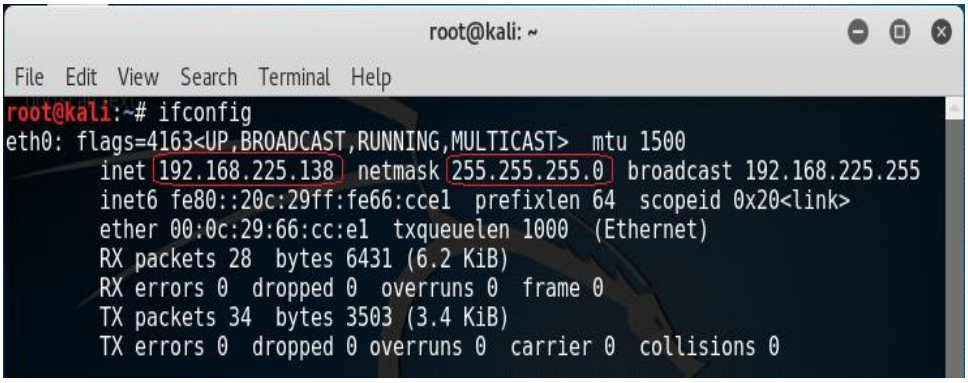
Hardware Requirements for This Lab:

- One virtual install of Kali Linux
- One virtual install of a Window or Linux target machine

Using Nmap Effectively

The usage of Nmap depends on the target machine and the differences between simple (basic) scanning and advanced scanning. What follows are examples of some basic commands and their usage. In part II of this lab, the more advanced Nmap scanning techniques are shown.

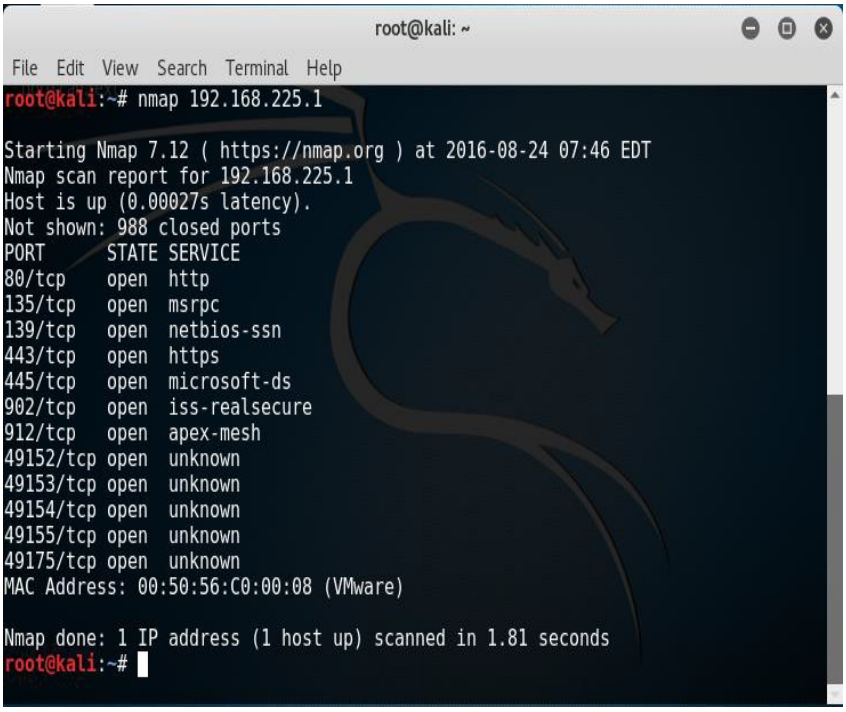
In this example, we are scanning a single IP address of 192.168.225.138. You can discover the IP address of your host machine by typing IFCONFIG at the terminal prompt. This will show you your assigned IP address. You can then substitute the default IP address used in the lab with the IP address of your machine for scanning a single host.



Answer the following questions:

- **What is your IP address?** In the above image, eth0 interface has been assigned the IP address of 192.168.225.138. This is my IP address; yours will differ!
- **What is the network portion of your IP?** For any class C, it is the first three octets 192.168.225.x The x in the last octet represents the IP address assigned to the host. My Kali host was given 138 as it's the host IP.
- **What is your subnet mask?** My network is using a default class C network, so my subnet mask is 255.255.255.0. This reads to say the first three octets are all used up; the O (zero) in the last octet tells the network the entire 8 bits can be used to assign an IP to for up to 254 network devices.

To scan a single system, use **# nmap 192.168.225.1:**

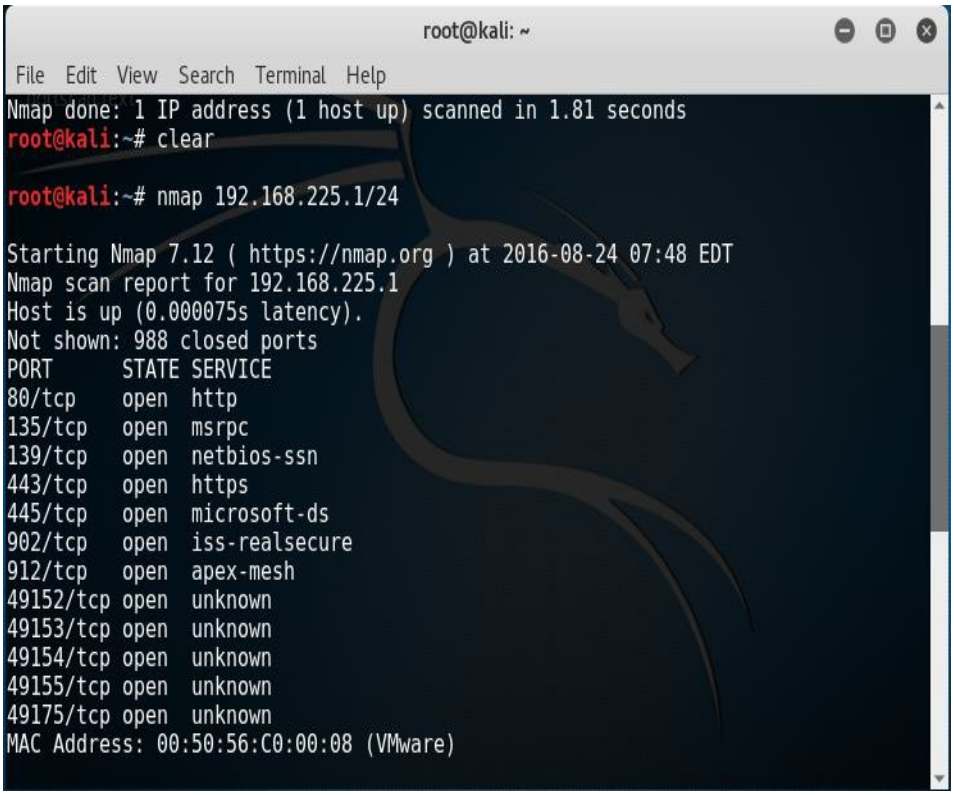


In this next example, we scan an entire class C subnet beginning with the first available IP of

192.168.1.225.1 The /24 is CIDR shorthand that tells Nmap to scan for all available IP addresses.

To scan the entire subnet use

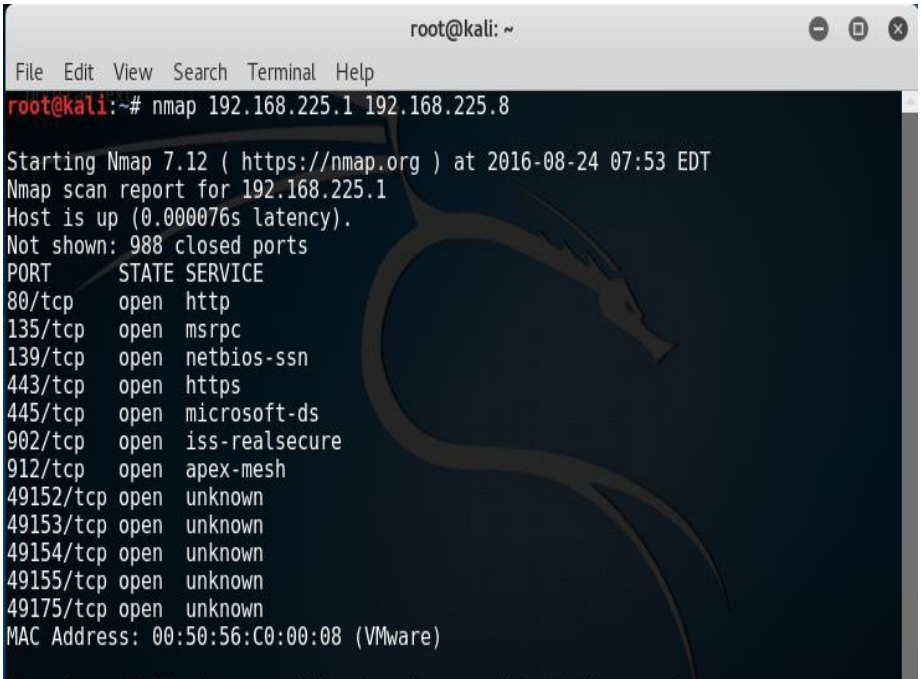
 # nmap 192.168.225.1/24:

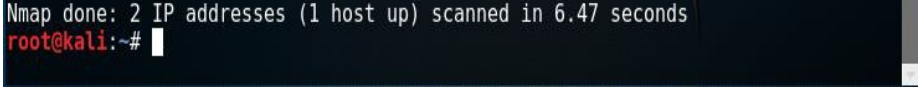


The previous command will discover all the hosts that reside on your network.

To scan multiple targets, separate each target with a single space:

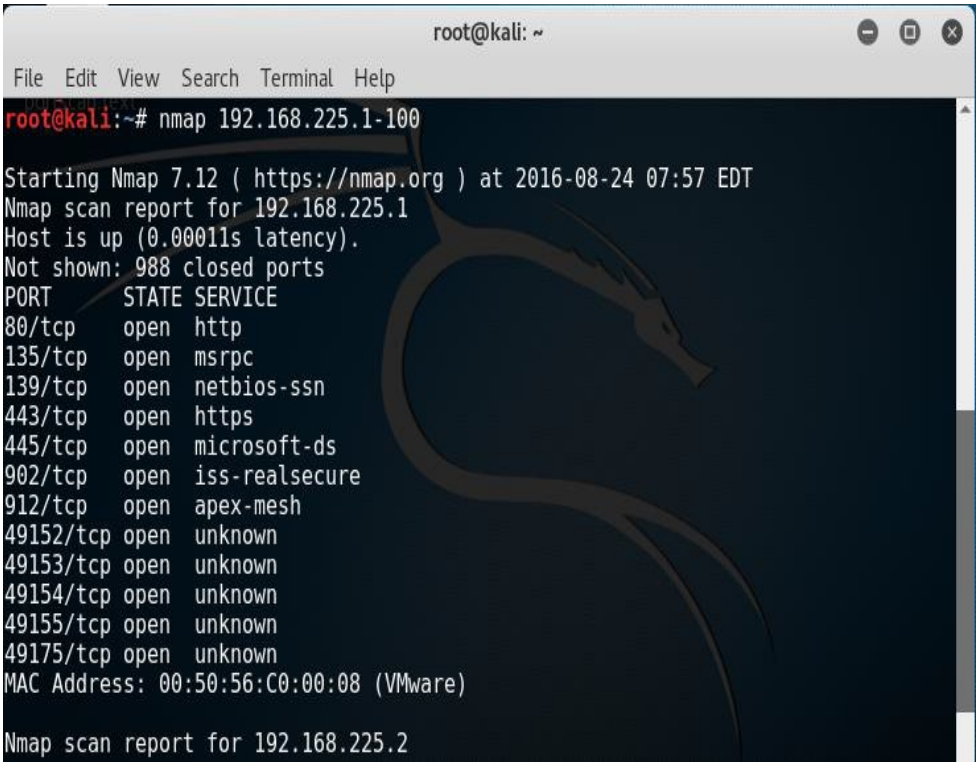
 # nmap 192.168.225.1 192.168.225.8





Using the previous command, I scan only two hosts with an IP address of 192.168.225.1 and 192.168.225. 8.

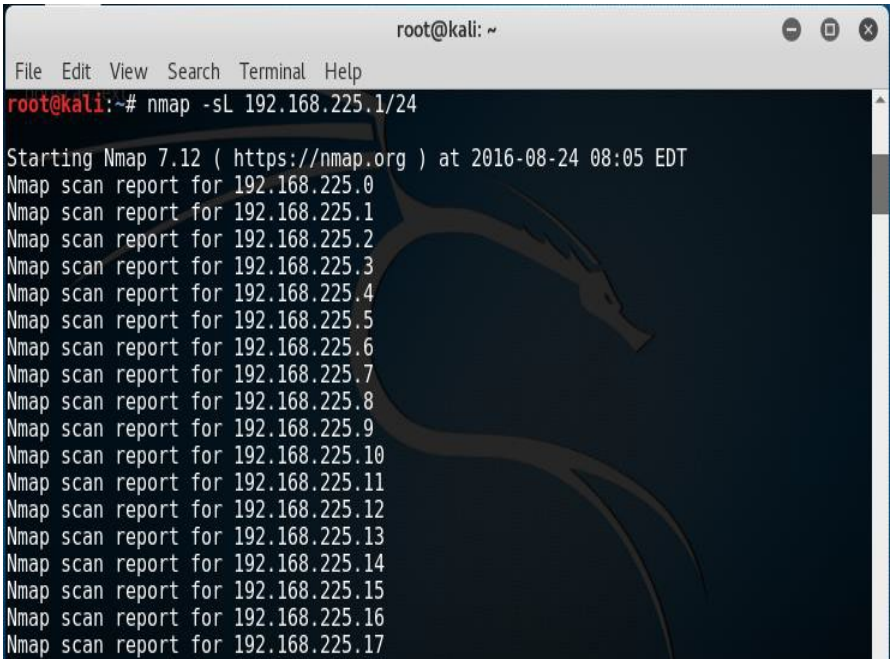
If you want to scan a range of IP addresses, but not the entire subnet, use this command:
 # nmap 192.168.225.1-100

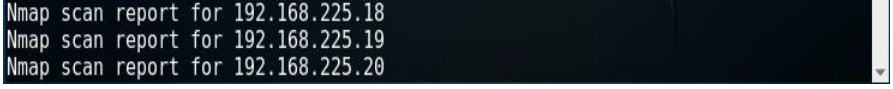


The previous command will scan for hosts with a starting IP address of 1 and end in 100.

If you want to see the list of all the hosts you are scanning, add the -sL parameter:

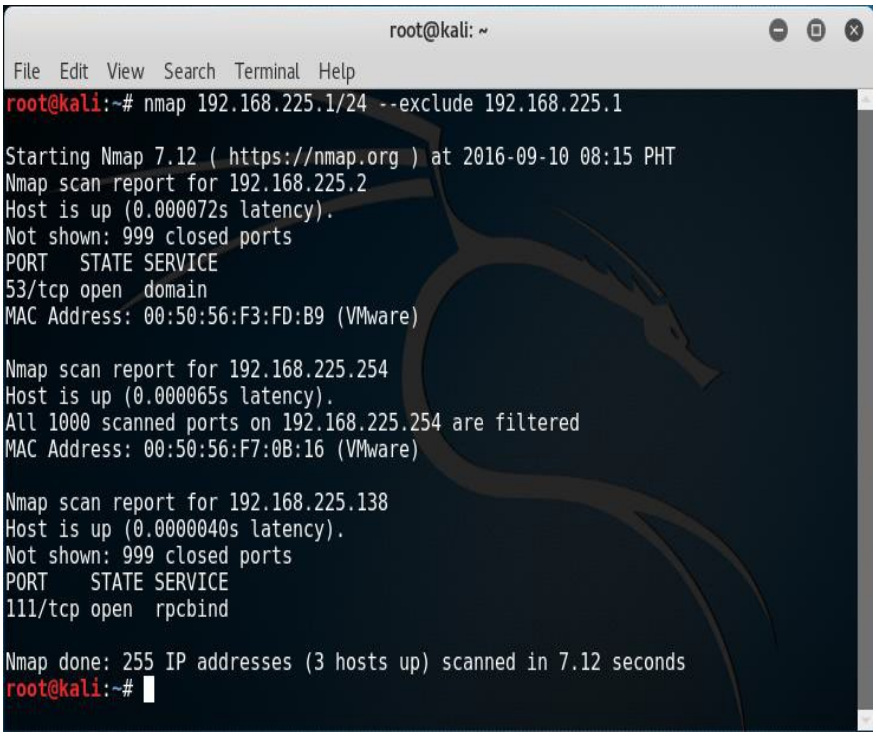
 # nmap -sL 192.168.225.1/24




Nmap scan report for 192.168.225.18
Nmap scan report for 192.168.225.19
Nmap scan report for 192.168.225.20

In some cases, we might need to exclude or exempt an IP from the scan. We can use the exclude parameter:

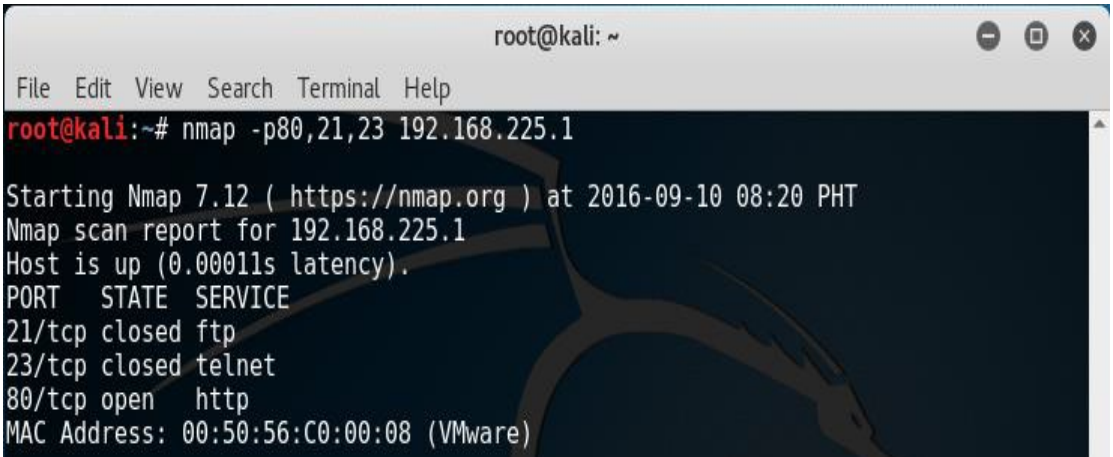
 # nmap 192.168.225.1/24 - -exclude 192.168.225.1

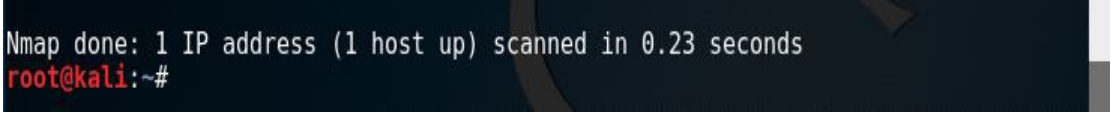


We scan for specific ports such as HTTP, FTP, and Telnet port using the Nmap parameter, -p:

 # nmap -p80,21,23 192.168.225.1

Replace the default IP address of 192.168.225.1 with the actual IP of any target machine, which is part of your network. Scan the IP for ports 80, 21, and 23.
Your results will differ from those in the following image. This is only an example of what you might see.




Nmap done: 1 IP address (1 host up) scanned in 0.23 seconds
root@kali:~#

The first part of the lab covered the basics of Nmap scanning. The second part will show us more advanced scanning techniques.

Part II

Nmap Scanning Techniques

In this section, we will discuss the more popular scanning techniques in detail.

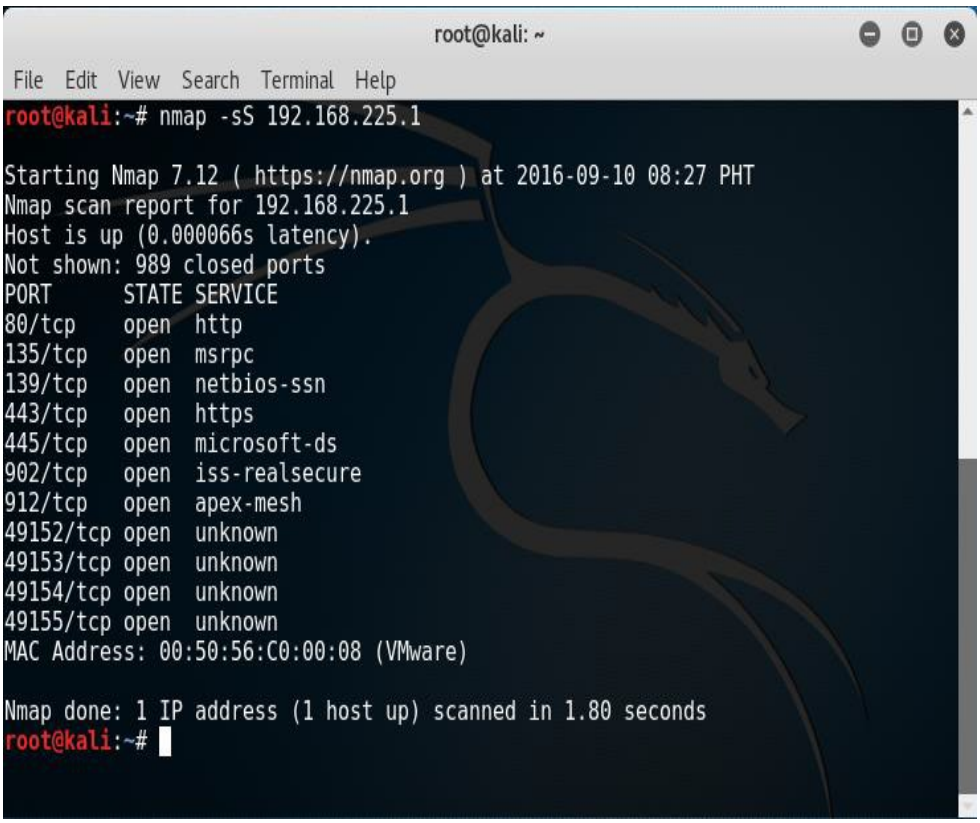
TCP SYN Scan (-sS)

It is a basic scan, and it is also called half-open scanning because this technique allows Nmap to get information from the remote host without the complete TCP handshake process. Nmap sends SYN packets to the destination, but it does not create any sessions. As a result, the target computer can't create any log of the interaction because no session was initiated, making this a feature of the TCP SYN scan.

If there is no scan type mentioned on the command, then a TCP SYN scan is used by default, but it requires the root/administrator privileges:

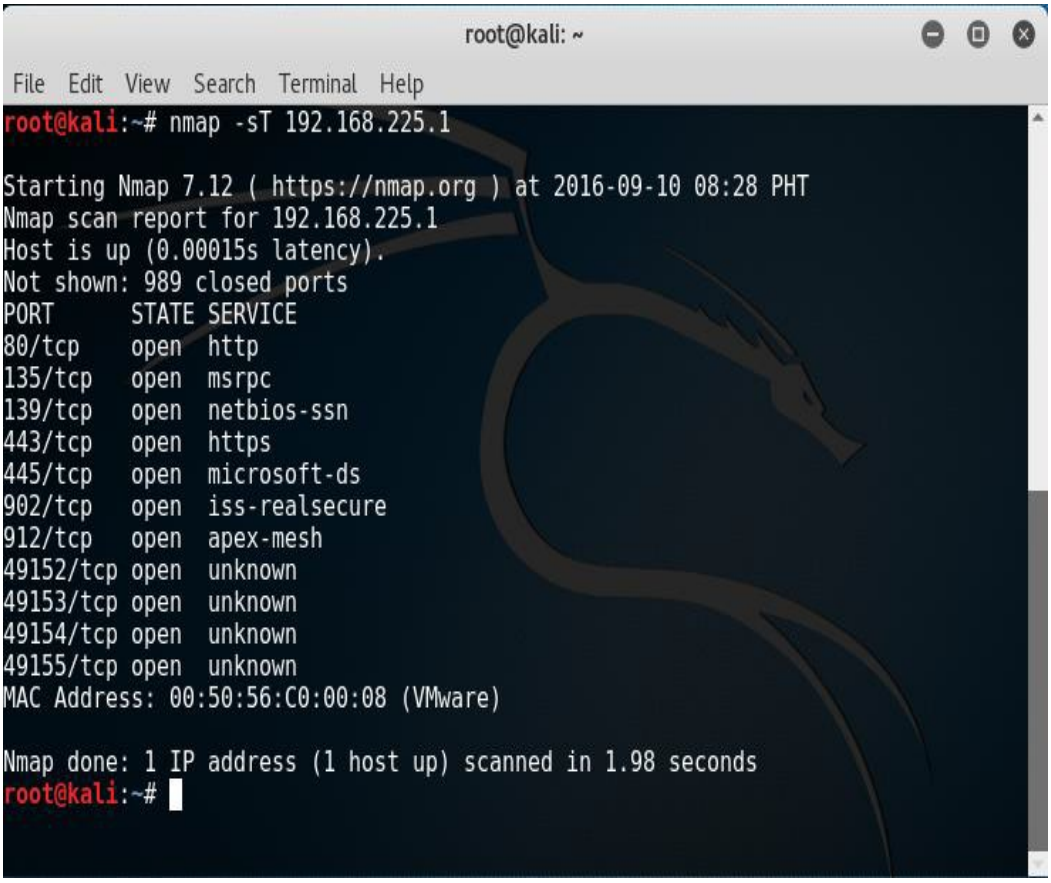
 # nmap -sS 192.168.225.1

(Replace the IP with that of a target machine running on your virtual network.)



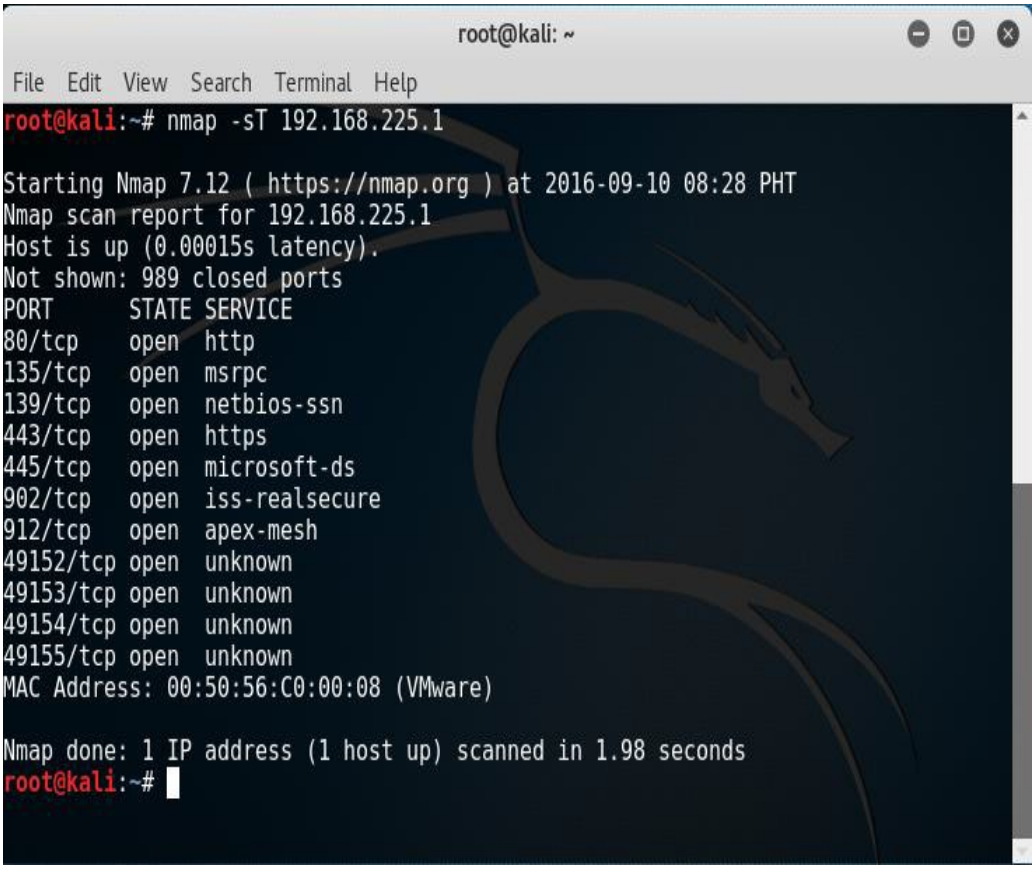
TCP connect() scan (-sT)

This the default scanning technique used, if and only if the SYN scan is not an option because the SYN scan requires root privilege. Unlike the TCP SYN scan, it completes the normal TCP three-way handshake process and requires the system to call connect(), which is a part of the operating system. Keep in mind that this technique is only applicable to find out the TCP ports, not the UDP ports.



 # nmap -sT 192.168.225.1

(Replace the IP with that of a target machine running on your virtual network.)

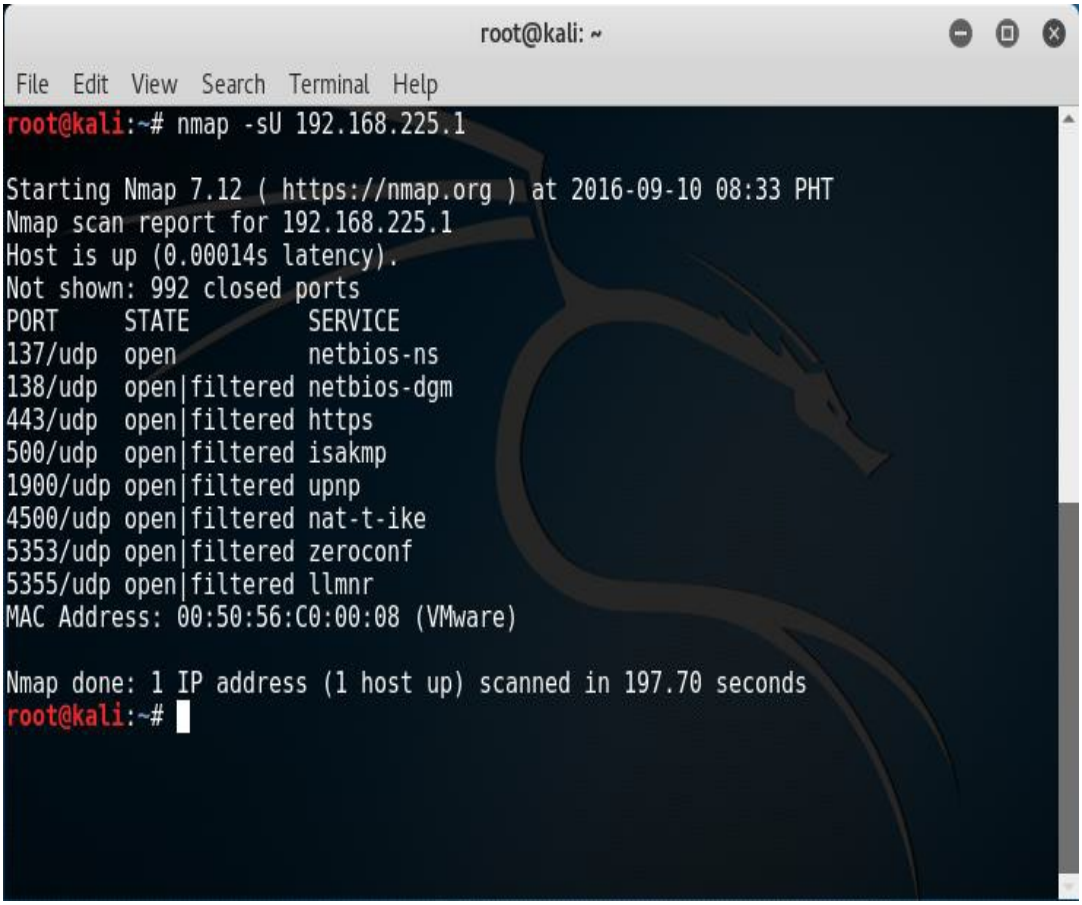


UDP Scan (-sU)

This technique scans for open UDP ports on the target machine. As we are using UDP, there is no three-way SYN handshake. But we can make the scanning more effective using -sS along with –sU. UDP scans send the UDP packets to the target machine, and wait for a response—if an error message arrives saying the ICMP is unreachable, then it means that the port is closed; but if it gets an appropriate response, then it means that the port is open.

 # nmap -sU 192.168.225.1

(Replace the IP with that of a target machine running on your virtual network.)

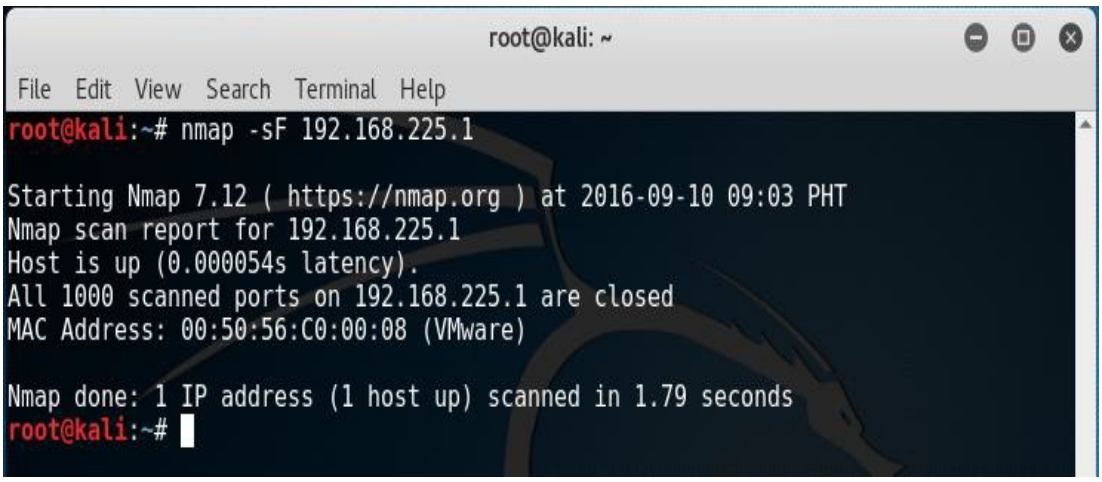


FIN Scan (-sF)

Sometimes a normal TCP SYN scan is not the best solution because of the firewall. IDS and IPS filtering can block the SYN packets. A FIN scan sends the packet with only a FIN flag, so it is not required to complete the TCP handshake.

 # nmap -sF 192.168.225.1

(Replace the IP with that of a target machine running on your virtual network.)



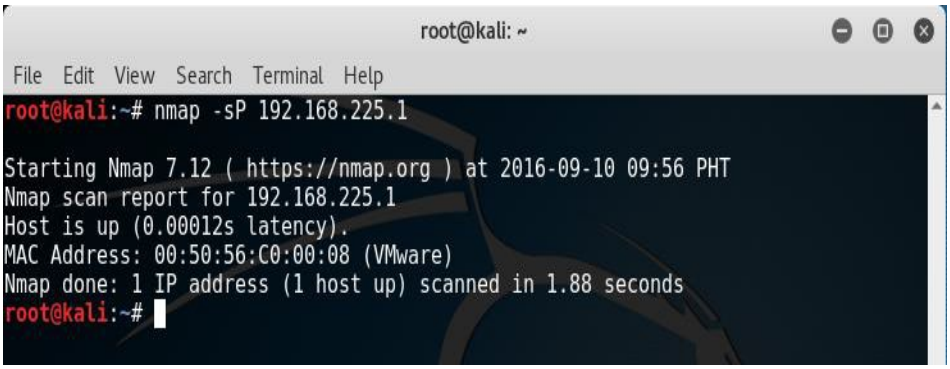
The target computer will not be able to create a log of this scan (again, an advantage of FIN). Just like a FIN scan, we can perform a xmas scan (-sX) and Null scan (-sN). The idea is the same, but there is a difference between each type of scan. For example, the FIN scan sends the packets containing only the FIN flag, whereas the Null scan does not send a bit on any ICMP packet. The xmas sends FIN, PSH, and URG flags.

Ping Scan (-sP)

Ping scanning is unlike the other scan techniques because it is only used to find out whether the host is alive or not, it is not used to discover open ports. Ping scans require root access so ICMP packets can be sent, but if the user does not have administrator privilege, then the ping scan uses connect() call.

 # nmap -sP 192.168.225.1

(Replace the IP with that of a target machine running on your virtual network.)

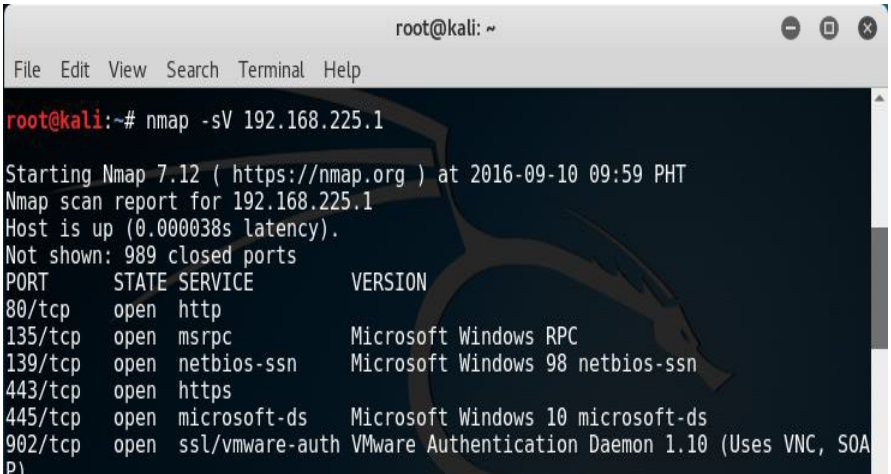


Version Detection (-sV)

Version detection is the technique used to find out what software version is running on the target computer and on the respective ports. It is unlike the other scanning techniques because it is not used to detect the open ports, but it requires information from open ports to detect the software version. In the first step of this scan technique, version detection uses the TCP SYN scan to find out which ports are open.

 # nmap -sV 192.168.225.1

(Replace the IP with that of a target machine running on your virtual network.)



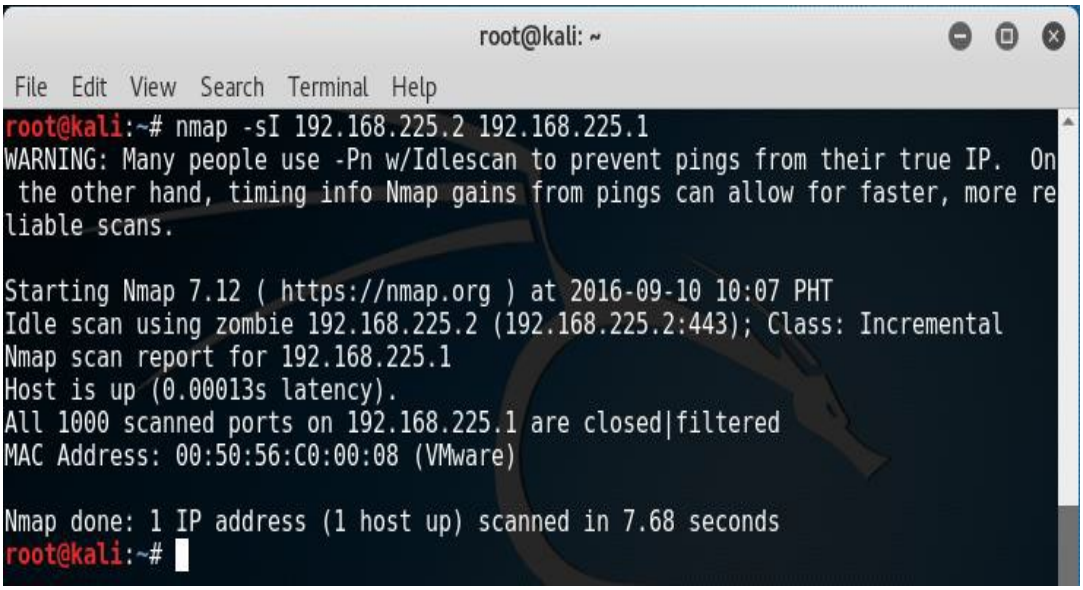

912/tcp open vmware-auth VMware Authentication Daemon 1.0 (Uses VNC, SOAP
)
49152/tcp open msrpc Microsoft Windows RPC
49153/tcp open msrpc Microsoft Windows RPC
49154/tcp open msrpc Microsoft Windows RPC
49155/tcp open msrpc Microsoft Windows RPC
2 services unrecognized despite returning data. If you know the service/version,
please submit the following fingerprints at https://nmap.org/cgi-bin/submit.cgi
?new-service :

Idle Scan (-sI)

Idle scan is a favorite technique of most hackers. This is an advanced scan that provides complete anonymity while scanning. In the idle scan, Nmap doesn't send the packets from your real IP address; instead of generating the packets from the attacker machine, Nmap uses another host from the target network to send the packets. Let's consider an example to understand the concept of idle scan:

 # nmap -sI 192.168.225.2 192.168.225.1

(Replace the IP with that of a target machine running on your virtual network.)



The idle scan technique is used to discover the open ports on 192.168.225.1 while it uses the zombie_host (192.168.225.2) to communicate with the target host. So, this is an ideal technique to scan a target computer anonymously.

There are many other scanning techniques available such as FTP bounce, fragmentation scan, IP protocol scan, and so on, but in this lab, you learned some of the more popular scanning techniques.

In this next section, you will learn Nmap's operating system (OS) detection and discovery techniques.

OS Detection

One of the most important features that Nmap has is the ability to detect remote operating systems and software. It is very helpful during a penetration test to know about the operating system and the software used by the remote computer because you can easily predict the known vulnerabilities from this information.

Nmap has a database called *nmap-os-db*, which contains information on more than 2,600 operating

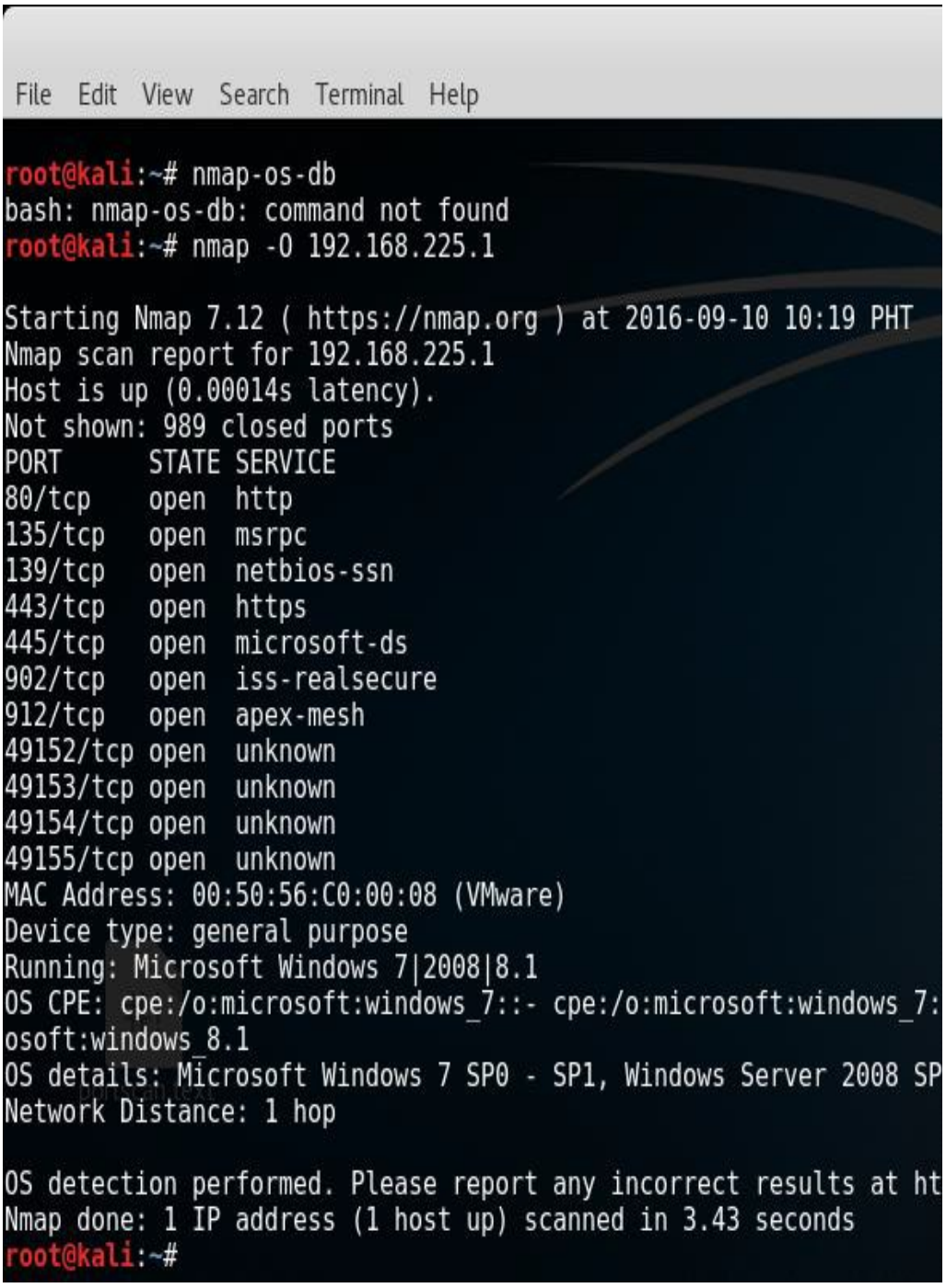
Nmap has a database called *nmap-os-db*, which contains information on more than 2,600 operating systems. Nmap sends TCP and UDP packets to the target machine and then it examines the response by comparing the result with the database. The Nmap operating system

discovery technique is slightly slower than the scanning techniques because OS detection involves the process of finding open ports.

The preceding example clearly demonstrates that Nmap first discovers the open ports, then it sends the packets to discover the remote operating system. The OS detection parameter is *-O* (capital O).

 # nmap -O 192.168.225.1

(Replace the IP with that of a target machine running on your virtual network.)



Your results will differ from those in the preceding image. This is only an example of what you

might see.

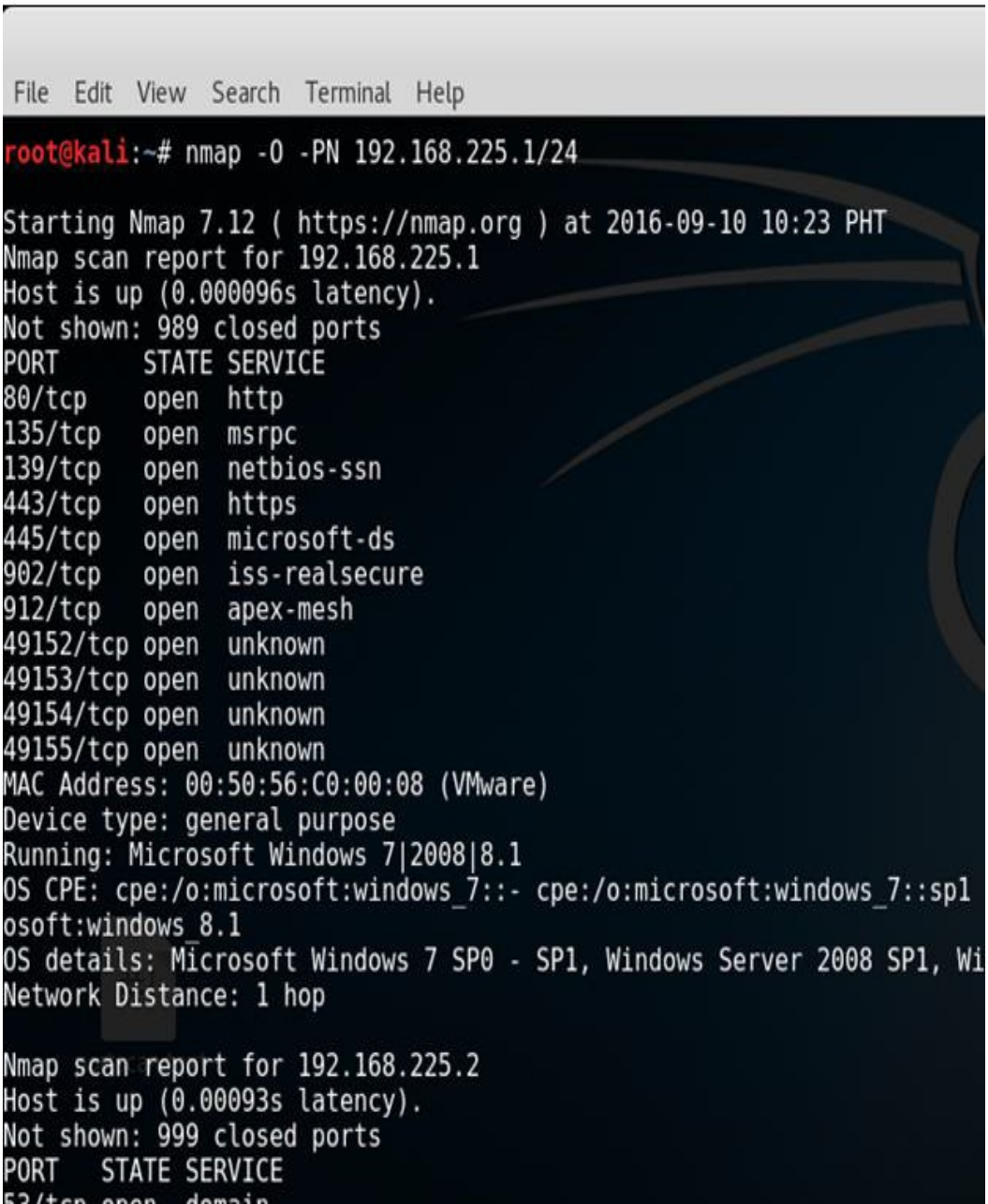
Nmap OS fingerprinting technique discovers the following:

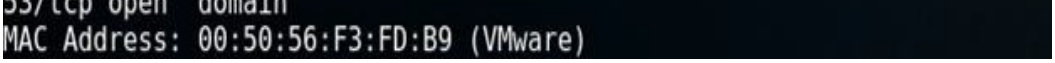
- Device type (router, workstation, and so on)
- Running (running operating system)
- OS details (the name and version of OS)
- Network distance (the distance in hops between the target and attacker)

Suppose the target machine has a firewall, IDS, and IPS all enabled. You can use the command ***-PN*** to ensure that you do not ping to find the remote operating system. The -PN tells Nmap not to ping the remote computer as sometimes firewalls block the request.

 # nmap -O -PN 192.168.225.1/24

(Replace the IP with that of a target machine running on your virtual network.)




53/tcp open domain
MAC Address: 00:50:56:F3:FD:B9 (VMware)

Your results will differ from those in preceding image. This is only an example of what you might see.

The command assumes every host on the network is alive so there is no need to send a ping request. We bypass the ping request and go straight to discovering the operating system.

The Nmap OS detection technique works on the basis of an open and closed port. If Nmap fails to discover the open and closed port, then it gives the following error:

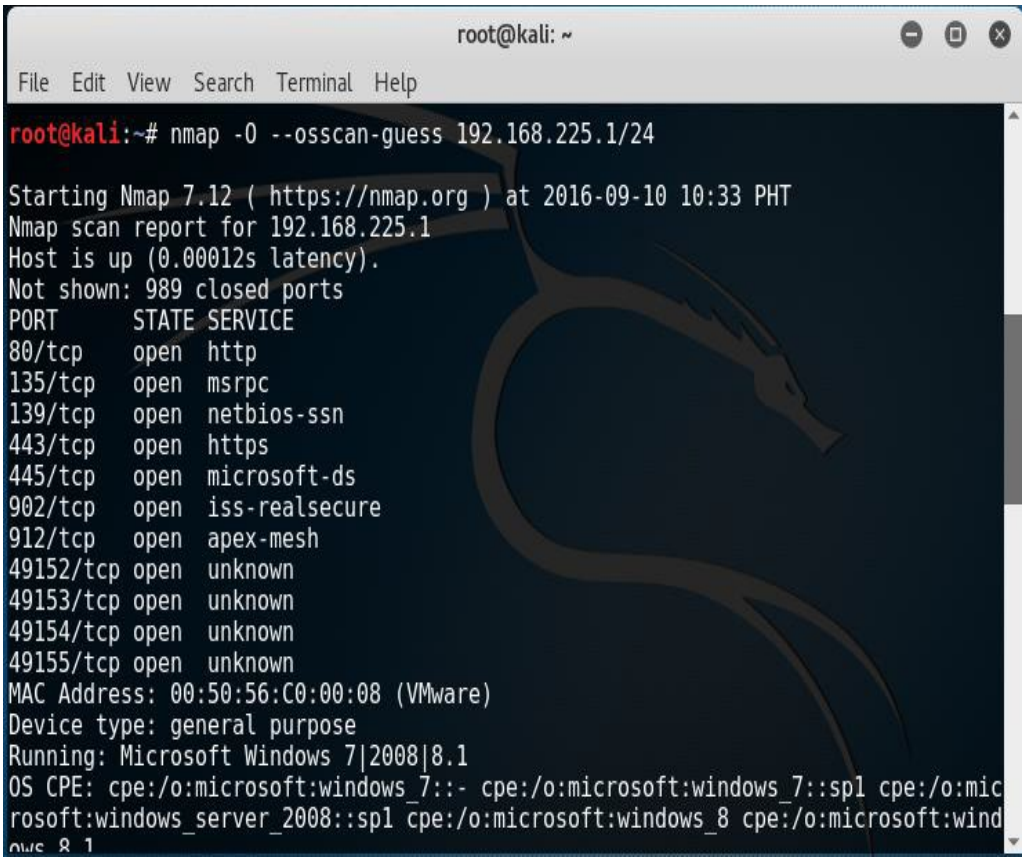
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port

This is an undesirable situation, and it is good to limit the operating system scans if Nmap is not sure about the OS. If Nmap is not sure about the OS, then there is no need to detect using ***–osscan_limit.***

If it is difficult for Nmap to detect the remote OS accurately, you have the option of using Nmap's guess feature; ***–osscan-guess*** finds the nearest match of the target operating system.

 # nmap -O --osscan-guess 192.168.225.1/24

(Replace the IP with that of a target machine running on your virtual network.)



Summary

Sometimes the best way to understand something is to see it in action. This lab included examples of Nmap used in most typical circumstances. Those new to Nmap should not expect to understand everything at once. This lab was a broad overview of Nmap.

As hackers, pentesters, security auditors, or network administrators, you learn to have a number of favorite Nmap commands to keep stored in our brain housing group. I have three or four favorites that cover roughly 99% of every discovery scan I need to perform. The trick with working in technology is really simple; you don't have to know all the answers, but you are expected to know how to find the answers.

End of the lab!