# CIT 285 - Lab #2: Network Scanning

## Introduction

In this lab, we will run our security tools on Kali, a Linux distribution designed for network security and penetration testing, to scan devices on various networks.

**References**

1. nmap documentation, http://nmap.org/docs.html
2. Secrets of Network Cartography, <http://www.networkuptime.com/nmap/index.shtml>

## 1: Ping Scans

We will run nmap in the terminal window. Find your IP address before proceeding. Commands require you to replace KALI\_IP with your actual Kali IP address.

1.1: Use a ping scan to check which of the 256 IP addresses in a /24 address space around your IP address are up. Since the returned output could be in the hundreds, we don’t want to manually count this output. In the second command, we’ll use grep with special options to do this for us by piping the output from nmap to grep.

**# nmap -sP KALI\_IP/24**

**# nmap -sP KALI\_IP/24 | grep -c ‘MAC’**

* How many IP addresses are up in this range?

There are 59 ip addresses up in this range

1.2: List the highest and lowest IP addresses that were shown to be up by the ping scan. The awk command is useful for parsing nmap output, as it can find the lines that you want using a pattern in //'s and then select the column of output that contains the information you want to retrieve.

**# nmap -sP KALI\_IP/24 | awk '/Nmap scan report/ {print $5}'**

**Highest = 10.2.240.203 (the last listed ip address is mine at 202, but the highest was 203)**

**Lowest = 10.2.240.2**

## 2: Port Scans

In this section, we will scan TCP ports in several different ways to gather information on an example target system. We will not scan UDP ports, as a UDP scan can take 15 to 20 minutes to complete.

2.1: What ports are open on the scanme.nmap.org test server?

*Your answer must include only the port numbers. Do not include other parts of nmap output.*

**# nmap -sT scanme.nmap.org**

22/tcp, 80/tcp, 646/tcp, 9929/tcp, 31337/tcp

2.2: Using a TCP SYN scan, provide a screenshot of other (if any) ports you found open on scanme.nmap.org?

**# nmap -sS scanme.nmap.org**

**A screenshot of a computer

Description automatically generated with medium confidence**

2.3: Looking at the output of the two scans outside the ports listed, what differences do you find between the TCP connect and SYN scans?

**The differences between a TCP and SYN is a TCP is a full connection compared to the SYN which is only half of the connection. The differences I see are how long it too, they look really similar.**

2.4: Some machines are behind a firewall, which filters connections to some ports, preventing nmap from receiving a response from those ports. Blocked ports may be listed as either “filtered” or “closed”.

To see an example of such a scan, perform a TCP connect scan on www.example.com. Your answer should be a screenshot of the returned port numbers for both closed and open ports. Do not include other parts of nmap output.

**# nmap -sT** [**www.example.com**](http://www.example.com)

Text

Description automatically generated

2.5: To determine why a scan returns the results that it does, use the --reason option. Explain the reasons that ports are listed as open, closed, or filtered in the scan of [www.example.com](http://www.example.com). Do a little outside research to answer this question fully.

**# nmap --reason -sT** [**www.example.com**](http://www.example.com)

**When a port is open this means that the connection of TCP is successful and is accepting TCP connections, this is the main goal when looking at ports. When a port is closed, this means that the application received and responded to this but the application is not accepting connections. When a port is filtered, this means that the Nmap cannot determine if the port is open due to such things like packet filtering preventing a connection. This is usually found in firewalls.**

2.6: To see every packet sent by a scan, use the --packet-trace option. We will save this output in a file for further analysis using I/O redirection. *STDERR are not redirected and will be returned in the terminal.*

**# nmap –-packet-trace -sS scanme.nmap.org > trace.out**

Open trace.out to view its content. What packet is sent first in the scan? Why do you think this particular packet is sent before the others?

Next, use the second command below to count how many packets are sent in total. The grep pattern ensures nmap content is not included in the count. *Remember when using the less command, use the ‘Enter’ key or spacebar to cycle/step through the output* *and ‘q’ to quit/exit out of less.*

**# cat trace.out | less**

**# less trace.out | grep –E "^SENT" | wc -l**

**The first scan is an echo request, this packet is sent before the rest because it has to find and establish a connection using request and response. After the second command there is 1005.**

2.7: The nmap scanner can return additional information, including service versions, OS identification, and tracerouting.

* The -A option will perform all these tests.
* The -T4 option tells nmap to use aggressive packet timing, which can be dangerous as it can cause some older machines to crash.
  + However, scanme.nmap.org is configured so that it will have no problems with the -T4 option. Even with the faster speed, this scan will take longer than previous ones due to the large number of tests performed.
* We add the -v option so that you can see the scan in progress. The less option makes the output manageable, remember to use spacebar or Enter to continue viewing nmap progress.

**# nmap -v -A -T4 scanme.nmap.org | less**

2.7: Based on the output of the scan, answer the following questions. *Note: Be sure to parse the output lines carefully.*

|  |  |
| --- | --- |
| **What is the server software name and version for each of the port found?** | 22 – tcpwrapped  80 – http and the version is apache/2.4.7  646- ldp  9929- nping-echo  31337- tcpwrapped |
| **What title would you see in the top of your web browser if you contacted the web server at scanme.org?** *Hint: think about the network protocol associated with Internet traffic* | Go ahead and ScanMe! |
| **How many network hops( i.e. different internet routers) did it take to reach the scanme.nmap.org server from your VM?** | There was 4 hops |

2.8: By default, nmap scans the most common 1000 ports.

* With the fast mode option (-F), nmap scans only 100 ports.
* The -sS options scans more ports, is fast, stealthy and reports port status of open, closed or filtered
* The -p option tells nmap to scan specific sets of ports, ranging from 1 port to the entire 65,536 possible ports.

Scanning all ports can take around 10 minutes over the Internet, so we will scan a local server (i.e. your Metasploit VM) for this question. *Be sure your Metasploit VM is powered on*.

**# time nmap -F -sS Metasploit2VM\_IP**

|  |  |
| --- | --- |
| **How many open ports did you find for this scan?** | 18 open ports |
| **How many wall clock(real) seconds did it take for the scan to complete?** | .223s |

**# time nmap -sS Metasploit2VM \_IP**

|  |  |
| --- | --- |
| **How many open ports did you find for this scan?** | 23 open ports |
| **How many wall clock(real) seconds did it take for the scan to complete?** | 2.752s |

**# time nmap -p0-65535 -sS Metasploit2VM\_IP**

|  |  |
| --- | --- |
| **How many open ports did you find for this scan?** | 30 open ports |
| **How many wall clock(real) seconds did it take for the scan to complete?** | 6.170s |

2.9: UDP scans are much slower than TCP scans due to the unreliability of UDP, so the scan in this question will require the longest amount of time. Could be up to 15 minutes. Provide a screenshot of open UDP ports.

**# nmap -v -sU scanme.nmap.org**

**A picture containing text, monitor, screenshot, screen

Description automatically generated**

## 3: Watching a port scan with Wireshark

**This section should be completed after the UDP scan in Question 2.9 completes.**

In a new terminal you’ll start Wireshark and begin capturing packets on eth0. Set a display filter of **ip.addr == KALI\_IP** in order to avoid counting irrelevant multicast or broadcast packets. We will use the **-Pn** option to prevent nmap from pinging the target machine before beginning the port scan.

For the remaining questions, think about what is happening when nmap scans a remote device. For example, what packets (i.e. protocol) are sent first? Think about how Wireshark works, what does Wireshark do in terms of a protocol analyzer.

3.1: How many packets are sent by nmap for a TCP connect scan of a single port? Does Wireshark see all the packets sent according to nmap? *Count only packets sent by nmap. Do not count packets sent by other machines or by other processes on your machine*.

**# wireshark &**

**# nmap –-packet-trace -p22 –Pn -n -sT scanme.nmap.org**

|  |  |  |
| --- | --- | --- |
| Number of packets sent by nmap during single port TCP scan | Number of packets captured by Wireshark during nmap scan | Does Wireshark see all the packets? yes/no |
| 3 | **4** | **no** |

3.2: Restart capturing packets to observe a SYN scan. How many packets are sent by nmap for a SYN scan of a single port? Does Wireshark see all the packets sent according to nmap? *Count only packets sent by nmap. Do not count packets sent by other machines or by other processes on your machine*.

**# nmap –-packet-trace -p22 –Pn -n -sS scanme.nmap.org**

|  |  |  |
| --- | --- | --- |
| Number of packets sent by nmap during single port SYN scan | Number of packets captured by Wireshark during nmap scan | Does Wireshark see all the packets? yes/no |
| 3 | 2 | yes |

## Submission

Upload a completed copy of this document to Canvas by the due date.