**Ethical Hacking Assignment 2**

**Network Enumeration and Port Scanning**

**Objective:** Determine which systems are accessible and the services/ports that are accessible with an active connection to target hosts. Rendering and discovering potential attack vectors in the system. Determine IP tables, port status, active services, banners, hostnames, settings, etc.

During this phase you will be conducting activities to scan the network devices performing Network Enumeration. Network Enumeration is the discovery of hosts/devices on a network, using discovery protocols such as ICMP and SNMP to gather information; they may also scan various ports on remote hosts for looking for well-known services in an attempt to further identify the function of a remote host. The next stage of enumeration is to fingerprint the Operating System of the remote host.

So the outcome of this phase is a list of IP on the network, which represents host, the open ports and services running on those ports and the operating system details of the hosts.

**Tools:** netdiscover, ping, fping, hping3, Nmap/Zenmap, OpenVas, Nessus and CVE information https://CVEDetails.com and National Vulnerability Database https://nvd.nist.gov/vuln/search.. (there are many more tools, but these will be the focus of this lab)

**Please make sure to get target IP from lab manager, to access the VCL.**

**Network Assignment:**

1. **Determine subnet to best conduct host discovery**

Determine for the subnet you are on the following items:

Subnet Mask: 255.255.255.0

Your Host IP­­­­­: 10.106.130.33

Range of IPs to scan: 10.106.130.1 – 10.106.130.254

What command did you use: ifconfig

With this information you can begin to test these host associated with your identified range are active (“alive”) or online.

Now, use **netdiscover** to determine which hosts are active in your subnet.

How many host did you discover: 96

Did you run the tool as passive or active? passive

Does this tool work across a router? No

Justify your answer: It only searches for hosts within the network.

1. **Host Discovery – using ping, fping, and hping3**

Using **ping** allows us to test if one host is active for every invocation of the **ping** command.

Enter ping <ip\_address> where ip\_address of the first host in your range

What are the values in the first response and define each value:

TTL:128

Time: 4.30ms

Seq: 1

From: 10.106.130.138

Bytes: 64 bytes of data

It can become very time consuming to perform a ping test for each host in a large subnet. In this case, it is best to conduct a ping sweep using a tool like **fping**.

**FPing** is a very handy tool for running ping sweeps on a network. There are a couple switches that you should be familiar with when using this tool. The first of which is **-a**, this switch is used to show only the active hosts in your output. This makes it much easier to read the final output from the sweep. Another important switch is that **-g**, this is used to specify that a range of IP addresses that you want to sweep. In addition, use the redirection operator > to save the large results in to a file, in this case a file called hosts.txt.

**fping -a -g 192.168.1.1 192.168.1.254 >hosts.txt**

This command will sweep the above IP address range and save the results to the hosts.txt file. You can view the hosts.txt file from any text editor or in Linux you can use the cat command in the terminal.

Use the command below in a terminal to immediately view the results:

**cat hosts.txt**

Run the command FPing on the range of host in your subnet, indicate how many are:

Active: 206 In Active: 48

The active IP in the hosts.txt file should be added to the list of available host network to targets.

**Note: The concern in using ping or fping is that both of these tools employ ICMP echo message request that are often block by firewall/switch/router. So, how can you test for active host and not get blocked, use Hping which employs TCP protocol to test host and cannot be blocked.**

**HPing**: To achieve greater control over host discovery and avoid detection or become more stealth use hping, which is a tool that allows you to control how the scanning takes place.

You might have heard about hping which is a command line tool and it is also called the TCP/IP analyzer, it is available on the Kali. <http://0daysecurity.com/articles/hping3_examples.html>

To send SYN packets: hping3 -S target

**hping3 -S ehacking.net**

If you want to do inverse mapping then RST packets are recommended: hping3 -R target

**hping3 -R ehacking.net**

**Inverse mapping** is a [stealth](http://searchsecurity.techtarget.com/definition/stealth)-approach [network scanning](http://searchmidmarketsecurity.techtarget.com/definition/network-scanning) method that gathers information about *inactive* [IP address](http://searchwindevelopment.techtarget.com/definition/IP-address)es on a network to try to determine which IP addresses are associated with *active* hosts. Typically, firewalls and routers do not respond to a [ping](http://searchnetworking.techtarget.com/definition/ping) response [packet](http://searchnetworking.techtarget.com/definition/packet) if the target address exists on the network, but respond with an [ICMP](http://searchnetworking.techtarget.com/definition/ICMP) (Internet Control Message Protocol) *host unreachable* signal if the target is not available. The absence of response allows the attacker to guess which IP addresses of a given address block map to live hosts. Inverse mapping may be a widely-used attack technique.

DDOS attack using hping

**sudo hping3 -i u1 -S -p <target port> <target ip> --rand-source**

Here -S indicates the SYN flag, -p means the desired port number, -i u1 means the delay of 1 micro second between each packet, --rand-source means to choose random source address.

**Using HPing,**

Run a Null, Xmas, and Ack scan of the same IP from when you used ping above.

**Show command for these scans:**

**Hping**

**Ttl = 128**

**Time = 3.0 ms**

**Seq = 0**

**From = 10.106.130.138**

Null: hping3 -c 2 -V -p 80 -s 5050 -Y

2 packages transmitted 2 received

Len=46 ttl=128 seq=0 rtt=0.4ms ack=287551059

Len=46 ttl=128 seq=1 rtt=0.7ms ack=998200146

Xmas: hping3 -c 2 -V -p 80 -s 5050 -UPF

2 packages transmitted 2 received

Len=46 ttl=128 seq=0 rtt=3.4ms ack=671706323

Len=46 ttl=128 seq=1 rtt=0.6ms ack=414795804

Ack: hping3 -c 2 -V -p 80 -s 5050 -A

Len 46 ttl=128 seq=0 rtt=3.9ms ack=978940955

Len=46 ttl=128 seq=1 rtt=0.6ms ack=1835688974

Was there a difference between these scans and the ping scan?

There’s a massive time difference between these on the first sequence. Ping took 4.30 ms while hping3 at the most took 3.9 ms. Hping3 gave more information in a shorter time while ping kept repeating.

What is the main difference between these host scanners?

Hping3 has more information and options than ping. Which allows for more flexible and deeper scans.

**Note**: you can get help on using them by typing **command –h** (where command is the utility name like fping or nmap) or by typing **man command** (where command is the utility name like fping or nmap). Man allows accessing the command’s manual. Arrow keys allow navigating through the manual, and pressing **q** allows returning back to the command-line.

1. **Port/Services Discovery Nmap and Zenmap**

Identify Open ports, services running on those ports and the host operating systems using nmap/zenmap. <https://hackertarget.com/nmap-cheatsheet-a-quick-reference-guide/>

**Nmap** ("Network Mapper") is a free and open source utility for network exploration or security auditing. It uses raw IP packets to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics.

Nmap runs on all major computer operating systems, and official binary packages are available for Linux, Windows, and Mac OS X. In addition to the classic command-line Nmap executable, the Nmap suite includes an advanced GUI and results viewer (Zenmap).

**Network mapping and IP Packet crafting using Nmap**

1. Type **nmap -h**, you should get the syntax for nmap along with its options.
2. To send a SYN packet to the computer that hosts Google web service and scan that computer in order to know what UDP/TCP ports are open, type **nmap –sS <target\_IP)** followed by the ENTER key. Note: Depending on how much time has passed since you used the sudo command, you may need to type sudo before the command you just typed. If needed, type the right command to get the result. (Why is this not a problem with Kali ?) Kali is faster and more reliable with built in tools that allow for easy scanning.

Command to scan Google host: nmap google.com

1. Show and perform the following scans on the given target IP address. You may need to use the –v option to interactively see what is going on.

On Kali Machine with XP IP address

1 host

1. total ports
   1. **TCP connect scan: used: nmap -v -sT**

**994 ports closed**

**Ports: 135/tcp State: open**

**Ports: 139/tcp State: open**

**Ports: 445/tcp State: open**

**Ports: 1025/tcp State: open**

**Ports: 3389/tcp State: open**

**Ports: 5000/tcp State: open**

**0.00012s latency**

* 1. **TCP SYN scan: used: nmap -v -sS**

**994 ports closed**

**Ports: 135/tcp State: open**

**Ports: 139/tcp State: open**

**Ports: 445/tcp State: open**

**Ports: 1025/tcp State: open**

**Ports: 3389/tcp State: open**

**Ports: 5000/tcp State: open**

**0.00025s latency**

* 1. **TCP FIN scan: used: nmap -v -sF**

**All 1000 ports closed**

**0.000025s latency**

* 1. **TCP Xmas Tree scan: used: nmap -v -sX**

**All 1000 ports closed**

**0.000038s latency**

* 1. **TCP Null scan: used: nmap -v -sN**

**All 1000 ports closed**

**0.000028s latency**

**Which of these is a stealth scan and why: TCP SYN scan is the stealth scan because it sends a SYN packet and waits and looks for a response. If something is sent back the port is open.**

4) nmap divides ports into 6 states: (define each)

* 1. Open: found from before: 6 (1025 445 135 3389 139 5000)

Define: The open port means it is accepting TCP connections, UDP datagrams, or SCTP associations on this port. This is the primary goal of port scanning.

* 1. Closed: found from before: All 1000 closed ports

Define: The closed port means it is receiving and responding to nmap probe packets but there is no application listening on it. Useful for later if they open.

* 1. Filtered: Found from kali ip: 2 (5800 and 5900)

Define: The filtered port means nmap cannot determine whether it is open because packet filtering prevents its probes from reaching the port which can be caused by a firewall or the router.

* 1. Unfiltered: Found from before: All 1000 ports are unfiltered.

Defined: The unfiltered port means the port is accessible but Nmap is unable to determine whether it is open or closed. This can be found with an ACK scan.

* 1. Open/Filtered: used: nmap -v -sU found from kali ip: 1 (68)

Defined: The open/filtered port means nmap is unable to determine whether a port is open or filtered. This is when open ports give no response.

* 1. Closed/Filtered: found from before: none

Define: The closed/filtered port means nmap is unable to determine whether a port is closed or filtered. It is only used for idle scan.

**4) Using ZenMap**

Now that you have obtained experience with nmap, let’s start using the GUI version of namp.

Run a full scan of your target system, recording the open ports, services running on those ports and the target’s OS.

Ports and Services

135 msrpc

139 netbios-ssn

445 microsoft-ds

1025 msrpc

3389 ms-wbt-server

5000 hppt-proxy

Target OS: Microsoft Windows 2000 SP0 – SP4 or Windows XP SP0 – SP1

Create a new Null scan profile and run it against the target IP, are there any differences?

This gave me no ports or services as well as no host details and took 14.20 seconds. It told me all 1000 ports are closed.

**Please perform these assignments outside of the lab**

**Vulnerability Identification and Scanning**

**5) Using CVEDetails.com**

Using the information from assignment 4, visit CVEDetails.com to learn about different vulnerabilities that exist for the machine scanned. Identify the top 5 CVE you would attempt to exploit.

CVE 1: CVE-2017-8487: Windows OLE in Windows XP a hacker to execute code when a victim opens a special file or program. CVSS score: 10.0

CVE 2: CVE-2014-4971: Microsoft Windows XP SP3 does not validate addresses in certain IRP handler routines. This allows local users to write data to arbitrary memory locations, and consequently gain privileges, via a crafted address in an IOCTL call, related to (1) the MQAC.sys driver in the MQ Access Control subsystem and (2) the BthPan.sys driver in the Bluetooth Personal Area Networking subsystem. CVSS score: 7.2

CVE 3: CVE-2014-0315: Untrusted search path vulnerability in Microsoft Windows XP SP2 and SP3 that allows local users to gain privileges via a Trojan horse cmd.exe file in the current working directory. CVSS score: 6.9

CVE 4: CVE-2014-0266: The XMLHTTP ActiveX controls in XML Core Services 3.0 in Microsoft Windows XP SP2 and SP3 allows remote attackers to bypass the Same Origin Policy via a web page that is visited in Internet Explorer, "MSXML Information Disclosure Vulnerability."

CVE 5: CVE-2014-0300: win32k.sys in the kernel-mode drivers in Microsoft Windows XP SP2 and SP3 allows local users to gain privileges via a special created application.

**6) Vulnerability Scanning – Nessus and OpenVas**

**Install Nessus personal on your own system and run a scan against your own system.**

Nessus tutorial: <https://www.youtube.com/watch?v=r_pDVhNoYr0>

Overview: <http://www.securitylearn.net/tag/nessus-tutorial/>

**Kali contains a comparable tool to Nessus called OpenVas, follow the video instructions to get it operational and then run a scan against your own system.**

OpenVas tutorial: <https://www.youtube.com/watch?v=qpdU1t2xyok>

Overview: <http://nest.unm.edu/files/4313/8379/8026/openvas.pdf>

Compare using Nessus and OpenVas, discuss the ease of use, information obtained and the advantages and disadvantage of using these vulnerability scanners.

I used both Nessus and OpenVas and out of both I can say I preferred Nessus just because of its easy web page like interface. Using this was as easy as just going on and putting host discovery under policies and letting it run its course. Its extremely customizable to stay organized and reports come in different colored charts to easy understand and visualize what is happening. It will also tell you the vulnerabilities you have. OpenVas on the other hand was opened through Kali command prompt which took a little longer than Nessus. The steps seemed more complicated and the web page like interface looked very messy and hard on the eyes. OpenVas on the other hand has more information from what I could tell available. Both have their pros and cons but personally Nessus was better for my experience.