Nmap-Part 1

(Network Mapper)

Running on console/command prompt

First part of two tier tutorial.Nmap is great security tool developed by “Fyodor”. Basically it was a *nix tool but now available on various platforms and with GUI as well. This tutorial is for newbie’s and skiddies who would like to learn the proper way of using it. Geeks can use it to brush up the things.

Let the IP address to be scanned is 192.168.0.1. Simply it can be done as:

nmap 192.168.0.1

Few default things have also been executed along with the above mentioned string. The actual string executed is:

#nmap –R –sS 192.168.0.1

Lets deal with “-R” here, will see –sS later on.

It’s a query to DNS server for reverse DNS name lookup i.e. requesting for some “name” attached with the specified IP address. It’s generally the case with servers. Hence if you don’t need the “name” desperately, avoid it using “-n” option.

#nmap –n 192.168.0.1 or #nmap –n –sS 192.168.0.1 (both are same)

‘-n’ disables Reverse DNS. Many DNS servers log name resolutions, so running an Nmap scan without disabling name resolution may cause Nmap station to appear in the DNS logs it attempts to resolve the name of every workstation it scans!

Disabling this option will speed up the scan manifold especially if you are scanning many machines simultaneously.

Now you may notice that Nmap doesn't do anything for a while and then suddenly it comes up with result. It actually does lot of work in that duration. To see all that you must use '-v' option, called as verbose.

#nmap –v –n 192.168.0.1

For more verbosity use 'v' twice

#nmap –vv –n 192.168.0.1

Scanning more than one machine

Ok, so up to here we were scanning one host only. What will you do to scan more than one host?

There are various ways of doing this. Let's consider few of them, rest you should be able to think of:

Suppose you have to scan 192.168.0.1, 192.168.0.2 and 192.168.0.3

nmap -vv –n 192.168.0.1,2,3 or

nmap –vv –n 192.168.0.1-3

generalizing further

#nmap –vv –n 192.168.0.1-3,6,12-20.

It will scan 1,2,3,6 and 12 to 20.

If you have to scan all the 254 machines:

nmap –vv –n 192.168.0.1-254 or

nmap –vv –n 192.168.0.* or

nmap –vv –n 192.168.0.1/24   (you should know subnetting for it)

nmap –vv –n 192.168.1-2.*. It will scan 192.168.1.0 to 192.168.2.255. It can also be written as

#nmap –vv –n 192.168.1,2.0-255

Hope you have enough brain to get these things.

Scanning specific ports:

Suppose you have to scan specific ports only and not the defaults ones. You should use '-p' for that

nmap –vv –p 80 192.168.0.1. It will scan port 80

nmap –vv –p 21,23,25,80-100 192.168.0.1

. It will scan port number 21, 23, 23 and 80 to 100.

nmap –vv –n –p 21,23,25 192.168.1-2.*

- Verbose mode (for interactive mode)
- Disabled reverse DNS lookup (speed up and doesn't let DNS server log anything)
- Scanning specific ports
- Scanning 192.168.1.0 to 192.168.2.254 machines.

Various Scanning options:

There are many scanning options available with Nmap. All have their advantages and disadvantages. You should use them according to your requirements.

· **-sS: SYN scanning**

TCP SYN scan gather information about open ports without completing the TCP handshake process. When an open port is identified, the TCP handshake is reset before it can be completed. This technique is often referred to as “half open” scanning.

It’s the default scanning technique if you are “root”. It’s the most common scan to use because it works on all networks, across all operating systems.

ADV:

The TCP SYN scan never actually creates a TCP session so isn’t logged by the destination host’s applications. And hence it’s a quiet scan.

DISADV:

You need privileged access to the system.

nmap –vv –n –sS 192.168.0.1

· **-sT: TCP connect scanning**

It performs the 3-way handshake.

ADV:

You don’t need to have privileged access.

DISADV:

Since it completes a TCP connection so apparent when application connection logs are examined.

I would suggest you to never ever use this scan.

nmap –vv –n –sT 192.168.0.1

· **-sF, -sX, -sN: FIN scan, Xmas tree scan, NULL scan**.

These are called “stealth” scans. They send a single frame to a TCP port without any TCP handshaking or additional packet transfers. They are more “stealth” than SYN scan and must be used if the remote machine is not a Windows-based machine. I’ll tell you why.

These scans operate by manipulating the bits of the TCP header. Nmap creates TCP headers that combine bit options that should never occur in the real world. These purposely mangled TCP header packets are thrown at a remote device, and nmap watches for the responses.

Window-based systems will reply with a RST frame for all queries, regardless of the status of the specific port that was queried.

ADV:

Since no TCP sessions are established, they are quiet stealthy.

DISADV:

Can’t be used against windows-based machine.

nmap –vv –n –sF 192.168.0.1

nmap –vv –n –sX 192.168.0.1

nmap –vv –n –sN 192.168.0.1

· **-sU: UDP scan**.

The only scan in the arsenal of Nmap to identify UDP ports.

nmap –vv –n –sU 192.168.0.1

· **-sO: Protocol scan**

Sometimes it has to be checked that what protocols the remote machine is running. It locates uncommon IP protocols that may be in use on the remote system. Hence it helps determining the type of remote device, i.e. is that router or printer or workstation etc.

DISADV:

This scan will appear on any network monitoring application that identifies the IP protocol types in use.

nmap –vv –n –sO 192.168.0.1.

· **-sR: RPC scan**.

It’s used to locate and identify RPC applications. It runs automatically during a version scan (-sV, explained later)

DISADV:

RPC scan opens application sessions and hence it will be logged.

nmap –vv –sR 192.168.0.1

· **-sV: Version scan**

The scans which we have seen by now give you the status of the port and the service running on them. For exploiting the service you need the exact version number of the service. Version scan gives you this.

DISADV:

It opens sessions with the remote applications, which will often display in an application's log file.

nmap –vv –sV 192.168.0.1

· **-sA: ACK scan**

Its quiet useful when there is some packet filtering device or firewall. It never locates an open port. It does the job of identifying ports that are filtered through a firewall. It doesn't open any application sessions and hence the conversation between nmap and the remote device is relatively simple.

DISADV:

It can only tell whether port is filtered or unfiltered. But can never definitively identify an open port.

nmap –vv –sA 192.168.0.1

· **-sI: Idle scan**

It's the stealthy most scan you can have. Tough to launch because you need a zombie for it. It would not be justice with this great scan to be described in just few lines. I would recommend you to read it in detail.

ADV:

You will never be caught.

DISADV:

Tough to launch as it’s not easy to find some zombie machine.

· **-sP: Ping scan**:

You must have heard of Ping sweep. It’s Nmap’s ping sweep.

nmap –vv –sP 192.168.0.10

will check whether this machine is up or not

nmap –vv –sP 192.168.0.*

will check the whole subnet (254) machines and will tell you which are up.

DISADV:

Ping scan will not interoperate with any other type of scan.

· **-sW: Window scan**

Forget it. As the number of operating systems vulnerable to its methodology is dwindling as operating systems are upgraded and patched.

· **-sL: List scan**

Would like to say only one line about it that you must use it if a separate application provides nmap with a list of IP addresses. Rest read yourself.

O/S fingerprinting and version detection

Ok, now you can use various scanning techniques to look for open/closed or filtered/unfiltered TCP as well as UDP ports. Don't you want to know the remote operating system running???

-O:

Operating system fingerprinting.

nmap –vv –O 192.168.0.1

It will tell you or at least tries its best to tell you the remote operating system along with the version it's using. It at least need one open and one close TCP port. In case it doesn't, it won't be able to give the accurate result. In that case you should use some third party tool.

DISADV:

A trained eye will quickly identify that someone is watching the network.

-sV:

Version detection

As has been explained it will help you know the version of the service running on the remote machine.

nmap –vv –sV 192.168.0.1

-A:

Named as Additional, Advanced, and Aggressive option. Its comprises of both the operating system fingerprinting process (-O) and the version scanning process (-sV).

i.e following two are same:

nmap –vv –sV –O 192.168.0.1 and

nmap –vv –A 192.168.01.

NMap tutorial for beginners-part 2

It's assumed that you are root. Many options won't work or better to say will switch to other kind of scans if you are not root. And sometimes it may even not give any warning before doing that. So beware!!! You may get logged.

Let's start it with various PING features available with Nmap.

PING is a necessary evil. I'll explain it later on.

Nmap Ping Methods:

First thing which should always be kept in mind is that Ping options are used to identify whether remote machine is up or not. Determining the open ports and services running on them is not the headache of Ping scan. Hence do your best using various Ping options to determine whether remote machine is up or down or being protected by some firewall.

Note: If user doesn't specifies a particular ping type, an ICMP Echo Request (-PE) followed by TCP ACK Ping (-PA) (by default on port 80 because most packet filters allow port 80) takes place. You can confirm it looking up the default ping options selected in the GUI Nmap.

Nmap provides various kinds of PING options and note that all of them start with the letter 'P'. Various combinations of these Ping options can be used in order to increase the chances of getting across packet filters and firewalls.

- **-PE ICMP Echo Request**

It's simply the ICMP Echo request and corresponding ICMP Echo reply packet and is best to determine the availability of machine. Drawback is that it's the most common protocol filtered by firewalls/packet filters. If you get response to this ping, it'll indicate that there is very less filtering between you and your destination.

- **-PA[port number] TCP ACK Ping**

Helpful to determine filtered/unfiltered ports, hence useful when there is some firewall protecting the machine.

#nmap –vv 192.168.0.1 –PA23,110

Here in this example, NMap will ping port 23 and 110 of the remote machine with ACK packets. If the remote machine is up or unfiltered, it will respond with RST packet. But in case it's down or ports are filtered, there will be no response and hence the scan will stop. Hence in order to get through firewall, try different ports.

If no port is specified, port number 80 will be pinged (which is generally the best one to ping as most packet filters allow traffic to port number 80).

These two above specified Ping scans run by default when you don't specify any kind of Ping scan.

- **-PS[port number] TCP SYN Ping**

Its functionality is same as SYN scan. Nmap machine sends SYN packet to remote machine. Open port will respond with ACK/SYN and closed will respond with RST. Hence can be used to determine whether remote machine is up or not. Ports can be specified, 80 is the default one.

#nmap –vv –n 192.168.0.1 –PS (will ping port number 80)

#nmap –vv –n 192.168.0.1 –PS23,110 (will ping 23 and 110 number ports)

-PU[port number] TCP UDP Ping

By default it sends UDP frames at port number 31338. UDP frames sent to closed ports responds with “ICMP port unreachable” message. If the remote port is open, it may or may not respond, because many UDP applications don’t send a response to any random incoming frame. Hence it should be tried to send the UDP frame to closed port. It heavily relies on ICMP packets, so if ICMP is filtered there may be no response to the UDP ping.

#nmap –vv –n 192.168.0.1 –PU (default port is 31338)

#nmap –vv –n 192.168.0.1 –PU<any port which you think would be closed>

-PP ICMP Timestamp Ping

ICMP Timestamp ping is used to allow two separate systems to coordinate their time-of-day clocks.

Avoid using it as NTP (Network Time Protocol) has replaced it. Hence Timestamp packets may raise eyebrows of trained eyes.

Moreover it doesn’t works properly when firewall is there as it relies heavily on ICMP.

- **-PM ICMP Address Mask Ping**

It operated by sending an ICMP address mask request to a remote device. Most modern operating systems and routers will not respond to this request, hence this ICMP ping type doesn’t work on most modern systems.

Hence forget it.

Conclusion:

If some firewall or packet filter is there on the remote device, better choice would be a non-ICMP based ping type.

-PO Don't Ping

If you know that the remote machine is up and running, you can use this option to remain a bit stealthier. Hence direct scanning of the target will start without pinging the machine. It should be used when using Decoys, otherwise ping packets will reach target from your machine only and from none of the decoys. Hence the purpose of using Decoys will loose its essence.

But as I told "Ping is a necessary evil", Nmap gather some important timing information from the ping process, so disabling the ping process will put nmap at a disadvantage when the scan begins. Actually it determines the accurate round-trip-time during ping.

Something about operating system fingerprinting

The usage of –O has been discussed in the 1st part of this tutorial. It gives the information of the operating system running on the remote machine which is must before launching some kind of exploit.

For this Nmap need at least one open and one closed port. If it doesn't get, it may not give the correct results.

+**--osscan_limit**

This option will abort OS fingerprinting if both open and closed ports are not available, hence will save a hell lot of time instead of getting incorrect results.

I personally feel that third party tools should also be used to get the correct results about the remote machines operating system.

-A (Additional, Advanced, and Aggressive)

Its combination of "-O" and "-sV" i.e. operating system fingerprinting and service version scan

#nmap –vv –n 192.168.0.1 –O –sV

is same as

#nmap –vv –n 192.168.0.1 -A

Inclusion and Exclusion of Hosts and Ports

Sometimes it may be the scenario that you don't want to scan particular IP address or range of IP addresses. E.g. Government IP's or IP's of routers and switches of your network etc.

So here we have few options for that:

Exclude Targets (--exclude <host 1, host 2, host 3….>)

The IP addresses specified will not be scanned by Nmap.

#nmap –vv –n –sS 192.168.0.1/24 --exclude 192.168.0.2-4, 192.168.0.7

This will scan the whole subnet except 192.168.0.2, 192.168.0.3, 192.168.0.4, and 192.168.0.7.

- **Exclude Targets in File (--excludefile <file name>)**

Here instead of specified the IP addresses which must not be scanned, user has to maintain a file including the list of IP address, one IP address per line.

#nmap –vv –n -sS 192.168.*.* --excludefile filename.txt

Content of filename.txt could be as following:

192.168.0.1-4 (exclude 192.168.0.1 à 192.168.0.4)

192.168.3-5.* (exclude 192.168.3.0 à 192.168.5.255)

192.168.6.* (exclude 192.168.6.0 à 192.168.6.255)

etc.

The benefit of –excludefile option is that a permanent exclusion file can be made including IP address of organizations/individuals whom you would never like to scan These IP address may

not be the part of current scan, but it won't harm making such a permanent exclusion file and upgrading it.

--excludefile and –exclude options can't be used on the same scan.

- **Read Targets from File (-iL <inputfilename>)**

Instead of supplying IP address at the command line of Nmap scan, a file can be maintained containing IP addresses separated by tabs, spaces, or by separate lines.

When this option is used, any IP address specified on the command line will be ignored without any warning message.

#nmap –vv –n –iL input.txt

#nmap –vv –n –iL input.txt 192.168.1.1 (Here 192.168.1.1 will be ignored)

If host exclusion options, --exclude or –excludefile, are used with –iL option, the excluded addresses will override any inclusions on the command line or file.

Scanning Random number of Targets

- **-iR <number of hosts to be scanned>**

You may be looking for just web server's or some other server's world wide, i.e. Random machines running a particular service on a particular port only.

E.g. telent at port number 23

SMTP at port number 25

Web server at port number 80 etc.

#nmap –vv –n –iR 100 –p 80

scan 100 random machines for port number 80

#nmap –vv –n –iR 0 –p 80 (that's zero and not capital 'O')

Scan "unlimited" number of machines for port number 80. So here you see, Nmap will scan thousands and thousands of machines. Scan won't begin to report any result until 500 hosts are identified. Hence run this type of scan with one of Nmap's logging option (will be discussed soon)

#nmap –sS –PS80 –iR 0 –p 80

It will run a TCP SYN scan using a SYN ping on port 80 to an unlimited number of random IP addresses. The SYN scan only scans port 80.

-iL, --exclude, --excludefile, none of them can be used with –iR option.

- **--randomize_hosts**

  #nmap –vv –randomize_hosts –p 80 192.168.*.*

As its clear from the scan command, nmap will randomize the hosts to be scanned.

-iL, --exclude, and –excludefile can be used with this option.

Groups of 2,048 hosts at a time are randomly chosen, and hence makes entire scan less conspicuous when examining traffic patterns.

V**arious Logging Options**

You may want to store the output of Nmap. Reasons may be any of the following:

1. You are scanning hundreds of machines, so don't want to stare at monitor for all the time.
2. You are starting the scan in night and would like to see the result in morning.
3. You may want to keep the records for future reference
4. You want to pause/stop Nmap and at later time would like to resume the scan.

…..and many more reasons could be there.

So here are the various logging options for NMap

- **Normal Format:**

-oN <log file name>

It saves a similar view of the output that's displayed on the screen during an nmap scan. No need to assign any extension to the output file. It will have .nmap extension.

- **XML Format**

-oX <log file name>

It presents the output of Nmap in very nice format in any browser. Actually Nmap includes an XSL file that translates the XML information into a viewable HTML format that can be displayed in any browser.

The output file will have .xml extension.

- **Grepable Format**

-oG <log file name>

The output file will have .gnmap extension.

- **All Formats**

-oA <base file name>

Will produce three output files, Normal, XML and Grepable.

Suppose the name of base file is target, so you will get the following three files:

1. target.nmap
2. target.xml
3. target.gnmap

- **Script Kiddie Format**

-oS <log file name>

Output would be in script Kiddie language

Resuming the scan

It's a good thing to do. You may be in the midst of a long scan when you have to turn off your computer or power failure etc. Can be interrupted using control-C key combination.

Suppose you are scanning a subnet of 100 machines and currently machine number 45 is getting scanned when you stop the scan. When you resume the scan next time, all the machines which

were scanned before machine number 45 won't be scanned again. But the scanning of machine 45 will start from scratch again as it was interrupted in between.

Hence do not use resuming option when you are scanning only one machine.

--resume <log file name>

This log file has to be either in Normal format (-oN) or Grepable (-oG) format. XML output won't work with it.

If a scan is interrupted that used the –randomize_hosts option, nmap has no method to recreate the same randomness that was used in the initial scan. Hence it may repeat some scan and may skip some of the hosts. So don't use it with –randomize_hosts option.

Few options which generally should always be included in all scans are:

-vv verbose mode

-n speeds up the scan and log is maintained on the DNS servers.

-oA to get output files so that scan can be resumed.

--excludefile It should be update with the most important IP addresses.