NMAP Tool for penetration Testing.

 |

Learning Selfy

Ahmed Almashani

**Introduction:**

The following Tutorial is about learning the basic levels of using the nmap tool for penetration testing. The goal here is to see how such tools can help discover a network. The goal here is to explore the various commands we can implement to see how it works….

**What is nmap tool?**

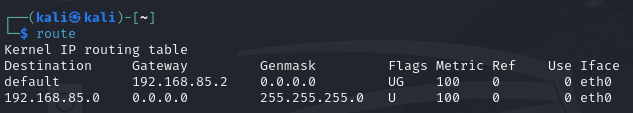
Nmap tool is a software that comes with almost all major distributions of linux, especially Kali, that can help investigate a network. For the purpose of this tutorial, we will simply stick to steps that can be followed to generate some answers and findings.

**Operating System used for this Tutorial:**

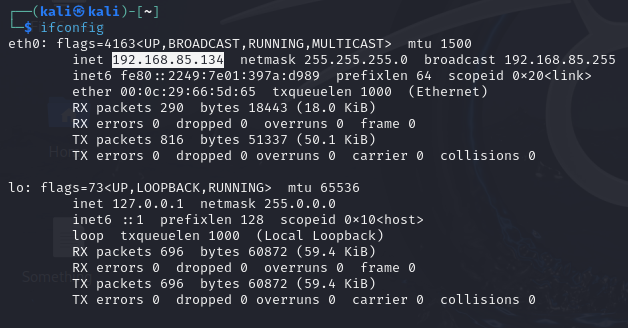
**Kali** Linux Operating System for nmap. It is among the most famous operating systems out there for penetration testing and it is, also, a reliable OS for such job.

**Starting the process:**

To start the process, we should issue the route command as it appears in the following picture:



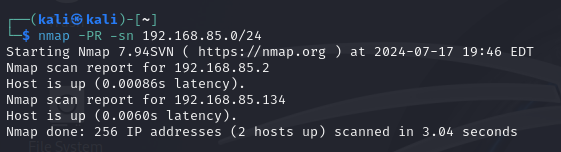
Issuing the command **route** displays the ip routing table and it helps in discovering the network. We can also issue the command **ifconfig** to display to know about our IP address as it appears in the following picture:



The highlighted IP address represents the VMware machine running kali and it is the only connected device to the network…

**Using nmap to scan Layer 2 (data-link layer):**

To scan the layer 2, we can issue the following command using the nmap tool:

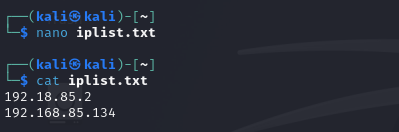


If you notice, we are using /24 to specify the network mask, but we can also use 192.168.85.0-255 instead. But the slash is more suitable and also, gives the same results. What the previous code does is that it sends a **broadcast** to all the devices within that range. The goal of his process is *to limit the targets of a network, in this case, the up and running hosts, instead of going out with a full blown broadcast that can waste time and also, alert the systems in the target network. For example, in this case, we found 2 IP addresses that are up and we can include them in a list for further testing and investigation. In other words, we are narrowing down our search.*

In the last line, we will see the nmap scanned 256 IP addresses and discovered 2 hosts up.

We can take the two hosts and save them into an iplist.txt file and use that list in the upcoming commands. The following picture shows how to create the iplist.txt:

Nano iplist.txt > enter 2 addresses (192.168.85.2 and 192.168.85.134) > save > exit

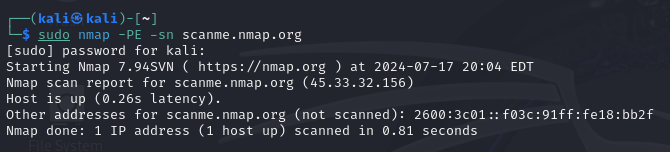


**Using nmap to scan Layer 3 (network Layer):**

To scan layer 3, we can test a website that is already available for such purpose and it is called:

Scanme.nmap.org

As it appears in the following picture:



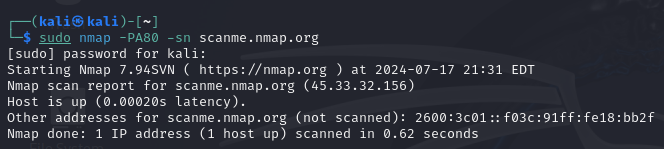
Please note that we used **sudo** to run this command as we need root privileges for it to work.

How it works:

*The command simply sends ICMP request to make sure there is an action on layer 3 and we got the results back. We know now the IP address, which is 45.33.32.156. There is a big chance you might be denied because of the target website is implementing a firewall that prevents such requests. And in this case, we will be using an alternative method, which is what we will be discussing below.*

**Using nmap to scan Layer 4(Transport Layer):**

The following picture illustrates that command used to perform layer 4 nmap on scanme.nmap.org website to see if we can get results:



From the first look, it seems we are getting the same results we got previously, but it is not. Here we are using Layer4 nmap testing whereas before, we used layer3 nmap testing.

***The******process*** *is sending acknowledgement to the host and it responded with TCP/IP packet indicating that we have received the packet and there is no open communication channel, and that means the host is up.*

**NMAP – PORT states:**

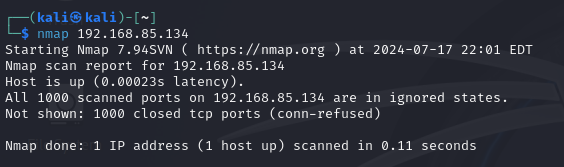
We have six types of port states in nmap:

1. **Open**: *an open port is one that is actively accepting TCP, UDP, or SCTP connections. Open ports are what interests us the most because they are the ones that are vulnerable to attacks. Open ports also show the available services on a network.*
2. **Closed**: *a port that receives and responds to nmap probe packets but there is no application listening on that port. Useful for identifying that the host exists and for OS detection.*
3. **Filtered**: *nmap can’t determine whether the port is open because packet filtering prevents its probes from reaching the port. Filtering could come from firewalls or router rules. Often little information is given from filtered ports during scans as the filters can drop the probes without responding or respond with useless error messages such as destination unreachable.*
4. **Unfiltered***: Port is accessible, but nmap doesn’t know if it’s open or closed. Only used in ACK scan which is used to map firewall rulesets. Other scan types can be used to identify whether the port is open.*
5. **Open/Filtered***: nmap is unable to determine between open and filtered. This happens when an open port give no response. No response could mean that the probe was dropped by a packet filter, or any response is blocked.*
6. **Closed/Filtered***: nmap is unable to determine whether port is closed or filtered.*

**Using nmap ports to scan:**

We have previously created a list of IPs to use and the reason why as mentioned previously is to keep a low-profile. Using the broadcast scanning can easily trigger firewalls to respond. The lower the list of targets, the low-profile we are using. Our list is named iplist.txt

But first let’s start with the most basic nmap scan:



The IP address belongs to the machine I am using now on Kali Linux, and the scan at the basic level usually runs on the most famous 1000 ports and as you can see, all the ports are closed, which is a good sign.

**Specifying a port on a target host to scan:**

To specify a target host on a specific port, we can simply issue the command: