

**Ethical Hacking**  
**Prof. Indranil Sengupta**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Kharagpur**

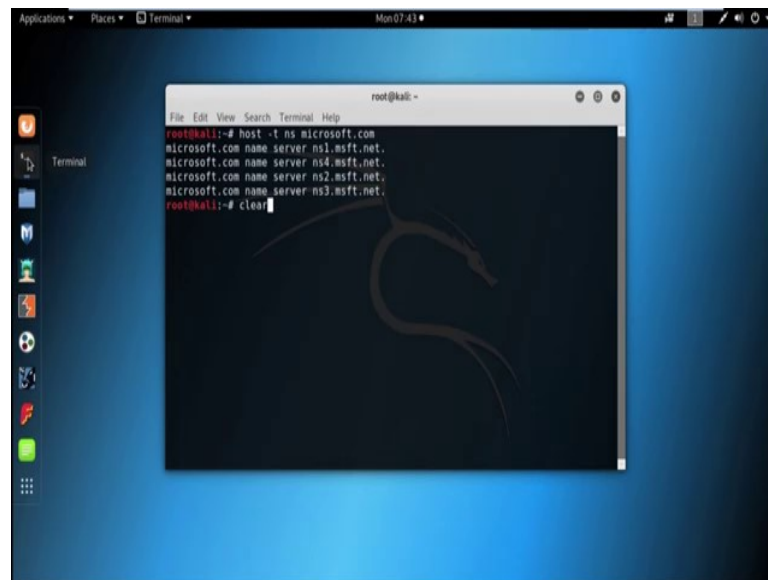
**Lecture - 18**  
**Demonstration Part III**

(Refer Slide Time: 00:15)



Then active reconnaissance, in active information gathering or in active reconnaissance, we gather information by directly communicating with effective. In active information gathering, we gather information about domain name system, open port, services, operating system etc. DNS enumeration information such as IP address, server name and often even server functionality can be discovered by DNS enumeration. We can interact with the DNS server using DNS client such as hosts and DNS lookup tool. Now, I will use the tool host for DNS enumeration.

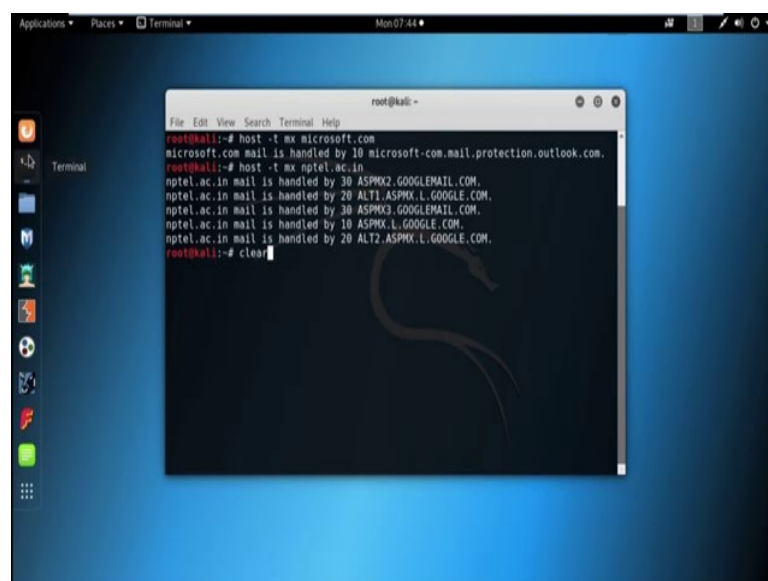
(Refer Slide Time: 01:21)

A terminal window on a Kali Linux desktop. The terminal shows the command `host -t ns microsoft.com` and its output, which lists three name servers for microsoft.com: ns1.msft.net, ns4.msft.net, ns2.msft.net, and ns3.msft.net. The user then types `clear` to clear the screen.

```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# host -t ns microsoft.com  
microsoft.com name server ns1.msft.net.  
microsoft.com name server ns4.msft.net.  
microsoft.com name server ns2.msft.net.  
microsoft.com name server ns3.msft.net.  
root@kali:~# clear
```

So, the command is *host*, then you need to specify the target by using *-t* and for name server we need to use the option *ns* followed by the domain name, suppose here we are using the domain name *microsoft.com*. So, we caught all the DNS server with the domain *microsoft.com*. Now, mail server enumeration, using mail server enumeration we can find information about all the mail server related with a particular domain, using the *host* command we can also enumerate the mail server; the command is like this.

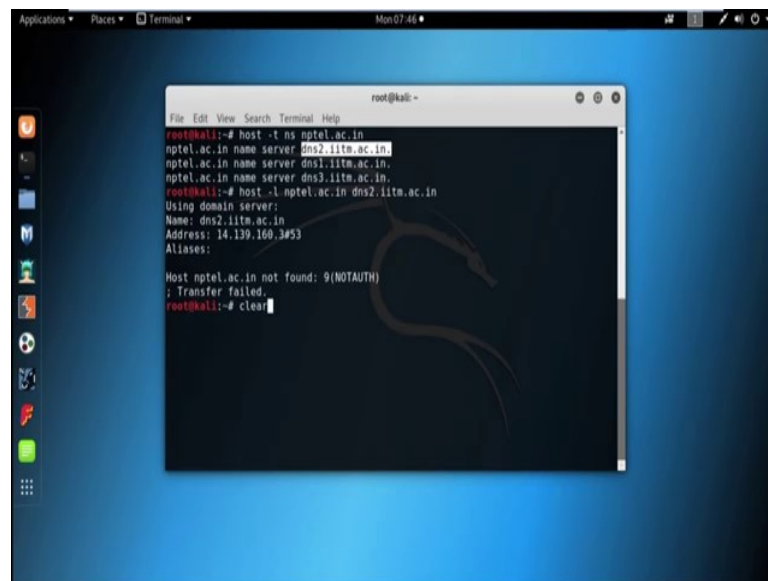
(Refer Slide Time: 02:25)

A terminal window on a Kali Linux desktop. The terminal shows two commands: `host -t mx microsoft.com` and `host -t mx npitel.ac.in`. The first command outputs mail servers for microsoft.com (10 microsoft-com.mail.protection.outlook.com). The second command outputs mail servers for npitel.ac.in (30 ASPMX2.GOOGLEMAIL.COM, 20 ALT1.ASPMX.L.GOOGLE.COM, 30 ASPMX3.GOOGLEMAIL.COM, 10 ASPMX.L.GOOGLE.COM, and 20 ALT2.ASPMX.L.GOOGLE.COM). The user then types `clear` to clear the screen.

```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# host -t mx microsoft.com  
microsoft.com mail is handled by 10 microsoft-com.mail.protection.outlook.com.  
root@kali:~# host -t mx npitel.ac.in  
npitel.ac.in mail is handled by 30 ASPMX2.GOOGLEMAIL.COM.  
npitel.ac.in mail is handled by 20 ALT1.ASPMX.L.GOOGLE.COM.  
npitel.ac.in mail is handled by 30 ASPMX3.GOOGLEMAIL.COM.  
npitel.ac.in mail is handled by 10 ASPMX.L.GOOGLE.COM.  
npitel.ac.in mail is handled by 20 ALT2.ASPMX.L.GOOGLE.COM.  
root@kali:~# clear
```

*host* then by using *-t* we need to specify the mail server *mx* that is mail exchange server followed by the domain name. Here we all again use the domain name *microsoft.com*, *microsoft.com* mail is handled by 10 *microsoft.com.mail.protection.outlook.com*. Now, suppose we are going to search the mail exchange server for the domain *nptel.ac.in*, *host -t mx nptel.ac.in*. Now, we got all the mail service for the domain *nptel.ac.in*, now DNS zone transfer.

(Refer Slide Time: 03:43)

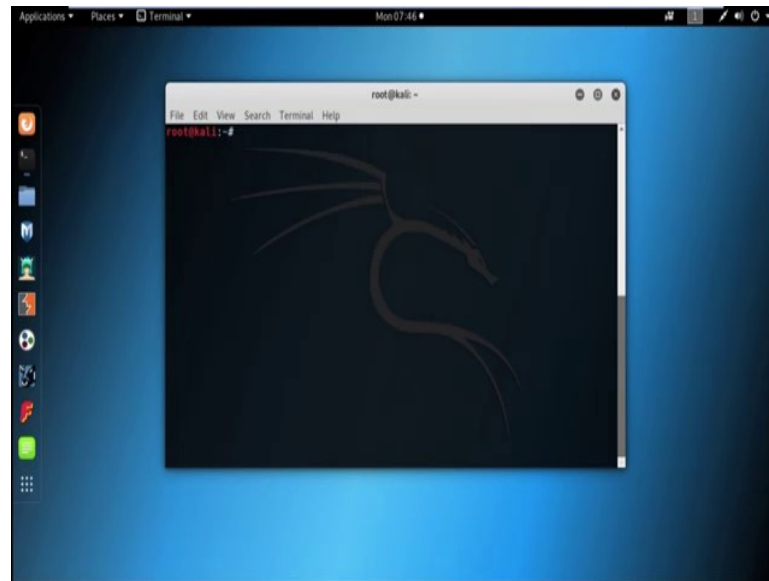


```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# host -t ns nptel.ac.in  
nptel.ac.in name server dns2.iitm.ac.in  
nptel.ac.in name server dns3.iitm.ac.in  
nptel.ac.in name server dns3.iitm.ac.in  
root@kali:~# host -l nptel.ac.in dns2.iitm.ac.in  
Using domain server:  
Name: dns2.iitm.ac.in  
Address: 14.139.160.3#53  
Aliases:  
Host nptel.ac.in not found: 9(NOTAUTH)  
; Transfer failed.  
root@kali:~# clear
```

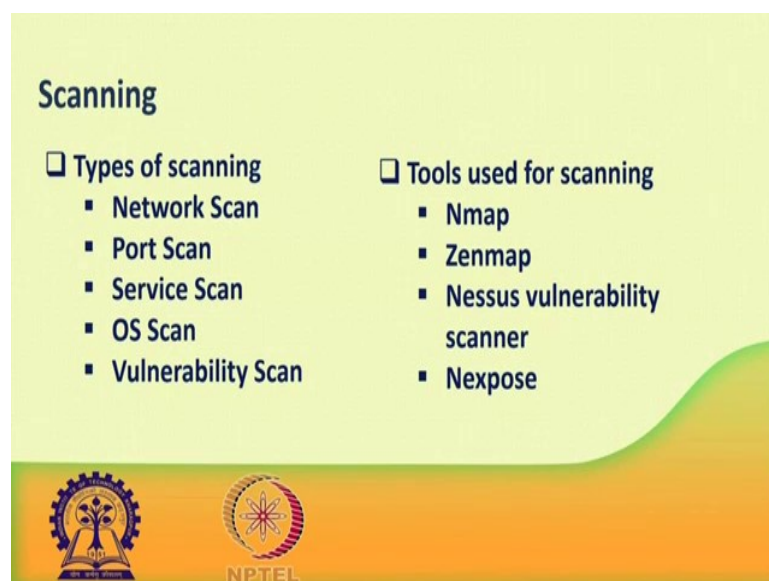
Zone file contained a list of all the DNS name configured for that particular zone, for this reason zone transfer should usually be limited to authorized secondary DNS service only. Unfortunately, many admin misconfigure that DNS service; as a result anyone asks for a copy of a DNS service zone file and receive it. Now, suppose I am searching for the DNS server of the domain *nptel.ac.in*, now we got three name server. Now, suppose I want to transfer the zone for the first DNS server; so, here I also use the command *host*, *host -l* then the domain name *nptel.ac.in* followed by the name server.

So, there is no permission for the zone transfer; so, that is why transfer failed.

(Refer Slide Time: 05:35)



(Refer Slide Time: 05:47)

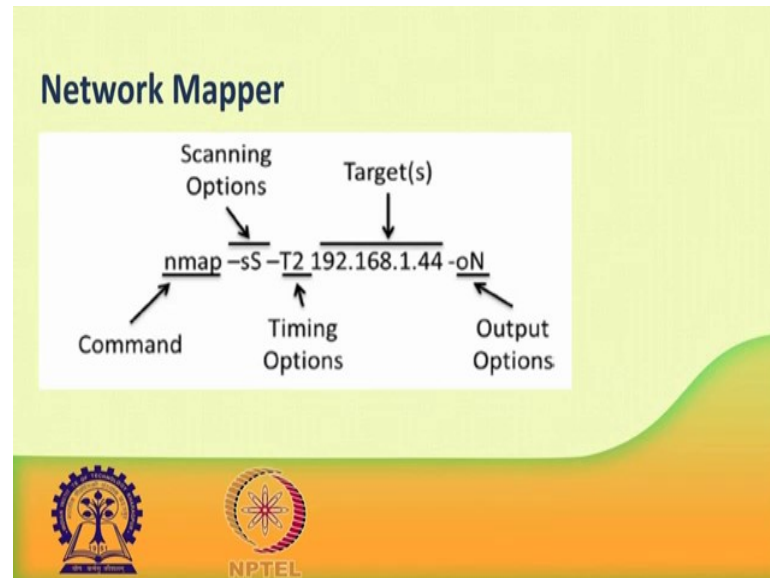


Scanning: in scanning the attacker begins to actively gather information from a target machine or network for vulnerabilities that can be exploited. There are different types of scanning at there like network scan, port scan version or service scan, OS scan, vulnerability scan etc.

Network scan: it basically detect the live host on the network, port scan detect the open port on the host, version a service scan detect the software and the version to the respective service running in any particular port. OS scan detect operating system,

vulnerability scan detect computers or computer systems or networks or applications for weakness.

(Refer Slide Time: 07:05)



Now, I am discussing some important scanning option; the small s, capital S, this is used for stealth scan. In this type of scan basically initiate a TCP connection with the target, but never complete the three way handshake that is why this is called stealth scan. The small s capital T used for TCP connect scan, the TCP connect scan can often be used to gather more information about the target than the stealth scan, as a full TCP connection is made with the targeted host.

The small s, capital U, UDP scan, the UDP scan access the UDP port on the target system; unlike scanning TCP port, UDP scan expect to receive replies back from system that have tested ports are closed, that is a state or ACK scan. The ACK scan is used to try to determine if a TCP port is filtered or unfiltered. Different timing templates are also there T0 that is paranoid, T1 sneaky, T2 polite, T3 normal, T4 aggressive and T5 insane; *nmap* offers the simpler approach with 6 timing templates.

You can specify them with the task capital T option and they are number from 0 to 5 or their net. The template name are paranoid, we use 0 for paranoid, sneaky we use 1, we used 2 for polite, we use 3 for normal, we use 4 for aggressive and we use 5 for insane scan. The first two are for ideas aversion, polite mode slows down the scan to use less

bandwidth and target machine resources, normal mode is the default and so, thus capital T3 does nothing.

Aggressive mode speed scans up by making the assumption that you are on a reasonably fast and reliable network. Finally, insane mode assumes that you are on an extraordinary fast network or are willing to sacrifice some accuracy for speed. This template allows the user to specify how aggressive they wish to be, while leaving *nmap* to pick the exact timing values, the templates also make some minor speed adjustment for which fine grained control option do not currently exist.

(Refer Slide Time: 10:43)

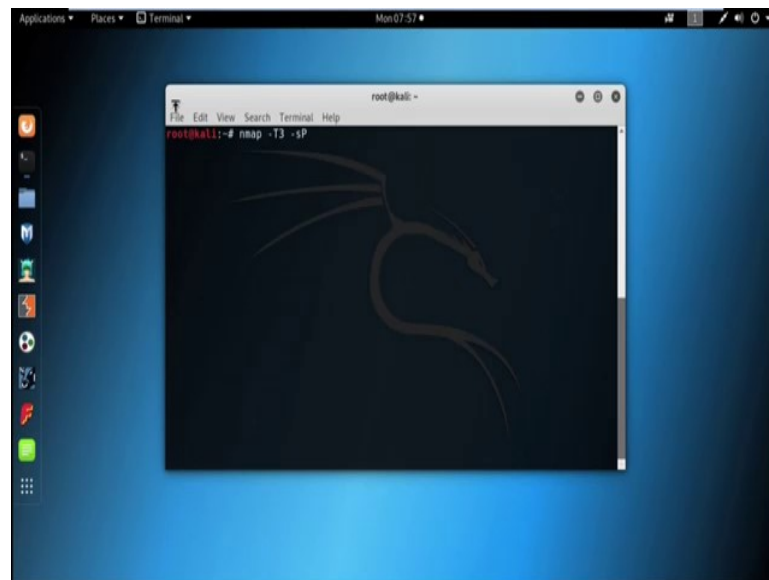
### Timing templates and their effects

	T0	T1	T2	T3	T4	T5
Name	Paranoid	Sneaky	Polite	Normal	Aggressive	Insane
min-rtt-timout	100	100	100	100	100	50
max-rtt-timout	300,000	15,000	10,000	10,000	1,250	300
initial-rtt-timout	300,000	15,000	1,000	1,000	500	250
max-retries	10	10	10	10	6	2
Initial (and minimum) scan delay (--scan-delay)	300,000	15,000	400	0	0	0
Maximum TCP scan delay	300,000	15,000	1,000	1,000	10	5
Maximum UDP scan delay	300,000	15,000	1,000	1,000	1,000	1,000
host-timout	0	0	0	0	0	900,000
min-parallelism	Dynamic, not affected by timing templates					
max-parallelism	1	1	1	Dynamic	Dynamic	Dynamic
min-hostgroup	Dynamic, not affected by timing templates					
max-hostgroup	Dynamic, not affected by timing templates					
min-rate	No minimum rate limit					
max-rate	No maximum rate limit					
defeat-rtt-timelimit	Not enabled by default					

Now, we are also providing the timing template and their effect and there are different output option are also there. We used as small o, capital N for normal output. The normal output option will create a text file that can be used to evaluate the scan result or use as input for other programs, thus small o capital X that is used for XML output.

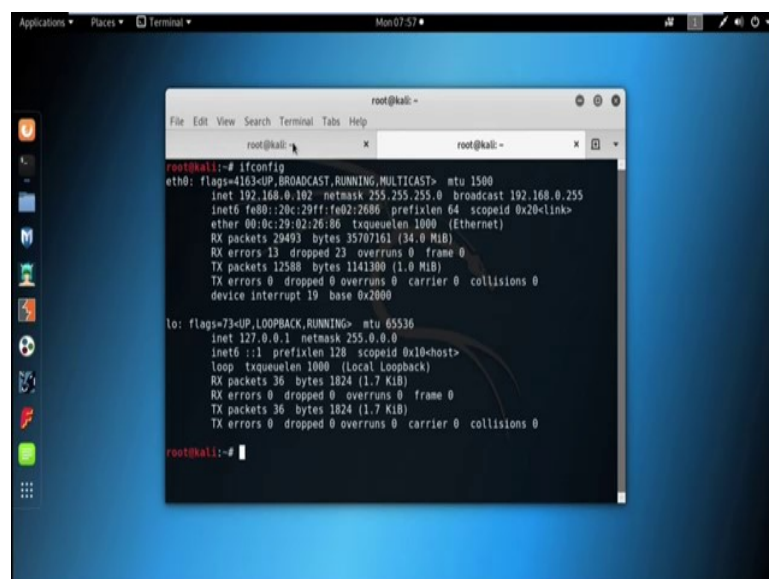
XML output can be used for input into a number of different applications for further processing or evaluation. The small o capital G grepable output, grepable output is often used by penetration tester to allow further investigation using tools like great. But, can also be searched using tools like AWK, ACD and DIFF, thus small o capital S script kiddies output while, not used for CDS penetration testing; the script kiddy output can be fun to use from time to time.

(Refer Slide Time: 12:27)



Now, I use the tool *nmap* for scanning. So, first type of scan is network scan; that means, need to find out all the live host in that network. So, first we need to use the command *nmap*, then we can use some timing option, then to find out all the live host in the network; we need to use the option the small s capital P and then the total range of the network.

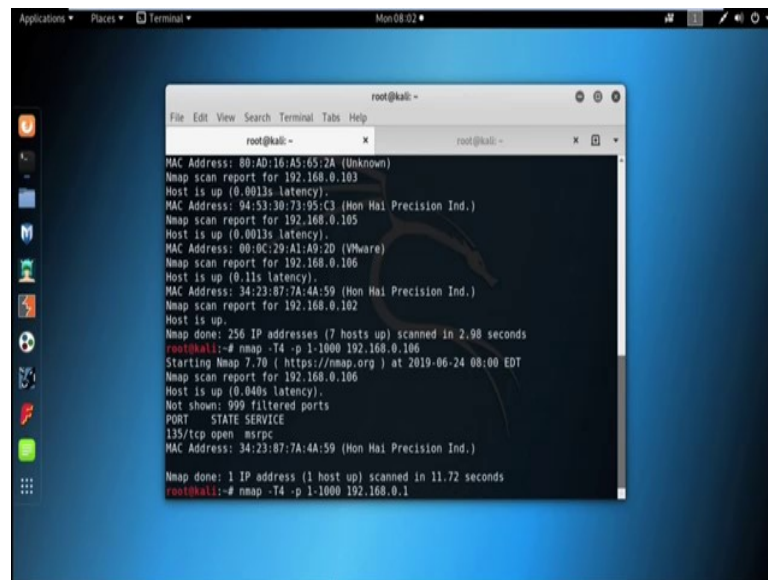
(Refer Slide Time: 13:21)



Now, finding out the IP address of my machine by using the command *ifconfig*. So, the IP address of my machine is 190.168.0.102 and the net mask is 255.255.255.0.



(Refer Slide Time: 13:41)



```
root@kali: ~  
File Edit View Search Terminal Tabs Help  
root@kali: ~  
MAC Address: 00:AD:16:A5:65:2A (Unknown)  
Nmap scan report for 192.168.0.103  
Host is up (0.0013s latency).  
MAC Address: 94:53:30:73:95:C3 (Hon Hai Precision Ind.)  
Nmap scan report for 192.168.0.105  
Host is up (0.0013s latency).  
MAC Address: 00:0C:29:A1:A9:2D (VMware)  
Nmap scan report for 192.168.0.106  
Host is up (0.11s latency).  
MAC Address: 34:23:87:7A:4A:59 (Hon Hai Precision Ind.)  
Nmap scan report for 192.168.0.102  
Host is up.  
Nmap done: 256 IP addresses (7 hosts up) scanned in 2.98 seconds  
root@kali:~# nmap -T4 -p 1-1000 192.168.0.106  
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-24 08:00 EDT  
Nmap scan report for 192.168.0.106  
Host is up (0.040s latency).  
Not shown: 999 filtered ports  
PORT      STATE SERVICE  
135/tcp   open  msrpc  
MAC Address: 34:23:87:7A:4A:59 (Hon Hai Precision Ind.)  
Nmap done: 1 IP address (1 host up) scanned in 11.72 seconds  
root@kali:~# nmap -T4 -p 1-1000 192.168.0.1
```

So, I can use the whole network range like this 192.168.0.0/255 or I can also use the total subnet like this. So, *nmap* started and it also give us the list of live host in this network. So, 192.168.0.1 is live and corresponding MAC address is also there, its a Tp link technologies. So, it is basically the router IP address. Now, 192.168.0.100 is also live, then 103, 105, 106, and 102, so, total 7 hosts are up.

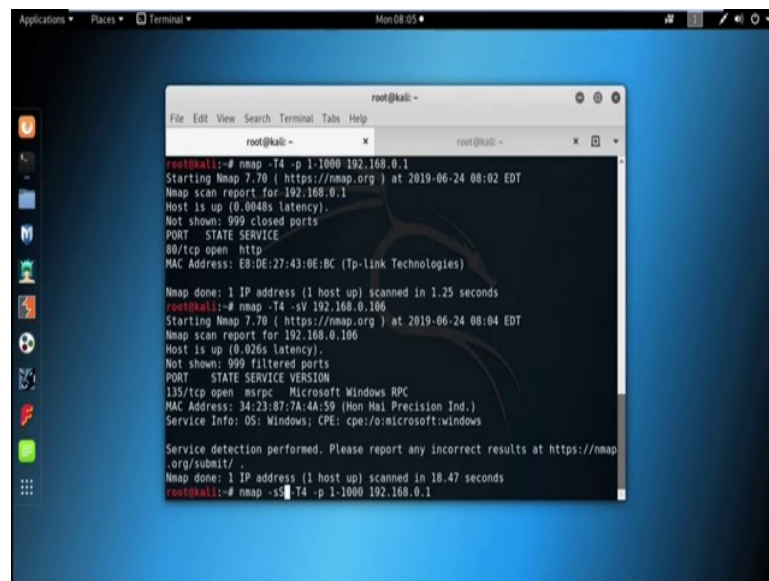
Now, next type of scan that is port scan. So, for port scan we need to use the option that small p followed by the port number or port name or port range also. So, suppose I want to scan the port from 1 to 1000, then I need to use the IP address. So, suppose I am searching means I am scanning the system with the IP address 192.168.0.106 ok. So, for first 1000 port 999 port are filtered and a single port 135 that is open and the service msrpc is running.

Now, the thing is that what are the filtered, unfiltered, open, closed port. So, open port basically an port that actively respond to an incoming connection. Closed port a closed port is a port and a target that actively responds to a prop, but does not have any service running on the port. Close port are commonly found on system where no firewall is in place to filter incoming traffic. Filtered, filtered port are that ports which are typically protected by a firewall of some sort that prevents *nmap* from determining whether or not the port is open or closed.



Unfiltered, an unfiltered port is a port that *nmap* can access, but is unable to determine whether it is open or closed. Open or filtered, an open filtered port is a port which *nmap* believes to be open or filtered, but cannot determine which exact state the port is actually in. Closed filtered, a closed filtered port is a port that *nmap* believes to be closed or filtered, but cannot determine which respective state the port is actually in. Now, I am scanning another host for first 1000 port. So now, the IP address target IP address is 192.168.0.1 ok.

(Refer Slide Time: 18:27)



```
root@kali:~# nmap -T4 -p 1-1000 192.168.0.1
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-24 08:02 EDT
Nmap scan report for 192.168.0.1
Host is up (0.0040s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
80/tcp    open  http
MAC Address: E8:DE:27:43:0E:BC (Tp-link Technologies)

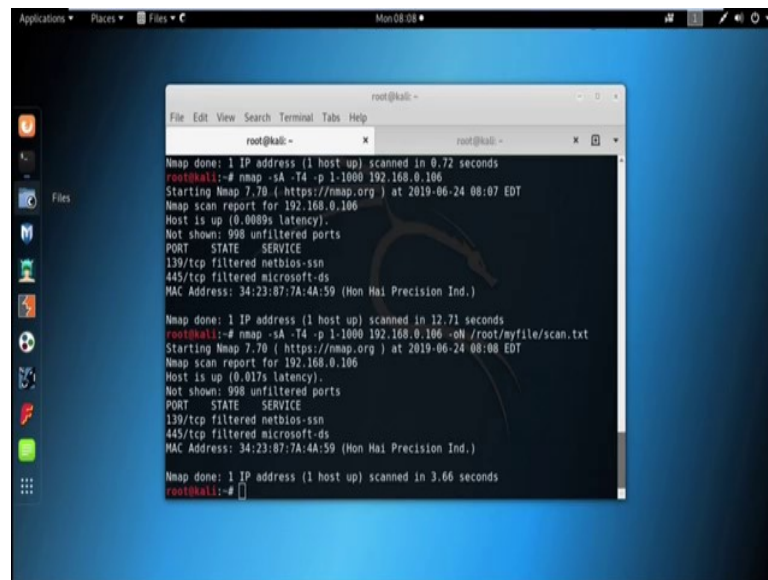
Nmap done: 1 IP address (1 host up) scanned in 1.25 seconds
root@kali:~# nmap -T4 -sV 192.168.0.106
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-24 08:04 EDT
Nmap scan report for 192.168.0.106
Host is up (0.026s latency).
Not shown: 999 filtered ports
PORT      STATE SERVICE VERSION
135/tcp   open  msrpc   Microsoft Windows RPC
MAC Address: 34:23:87:7A:4A:59 (Hon Hai Precision Ind.)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 18.47 seconds
root@kali:~# nmap -sS -T4 -p 1-1000 192.168.0.1
```

Port 80 is open for this host and http service is running, next service or version scan; using service or version scan we can find out the exact version of a service which is running in any particular port. Thus, small s capital V option is basically used for service or version scan, *nmap* then timing option *-T* capital 4. Then that is s capital V, then IP address 192.168.0.106.

Port 135 is open and msrpc service is running and version is Microsoft Windows RPC. Now, I am using different scanning option, suppose I want to scan the first 1000 port, first 1000 port using the stealth scan option. So, I need to use *nmap* then scanning option dash s capital S, then dash capital T4, then the small p followed by the port range and then IP address ok.

(Refer Slide Time: 20:57)



```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali: ~  
Nmap done: 1 IP address (1 host up) scanned in 0.72 seconds  
root@kali:~# nmap -sA -T4 -p 1-1000 192.168.0.106  
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-24 08:07 EDT  
Nmap scan report for 192.168.0.106  
Host is up (0.0089s latency).  
Not shown: 998 unfiltered ports  
PORT      STATE SERVICE  
139/tcp   filtered netbios-ssn  
445/tcp   filtered microsoft-ds  
MAC Address: 34:23:07:7A:4A:59 (Hon Hai Precision Ind.)  
  
Nmap done: 1 IP address (1 host up) scanned in 12.71 seconds  
root@kali:~# nmap -sA -T4 -p 1-1000 192.168.0.106 -oN /root/myfile/scan.txt  
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-24 08:08 EDT  
Nmap scan report for 192.168.0.106  
Host is up (0.017s latency).  
Not shown: 998 unfiltered ports  
PORT      STATE SERVICE  
139/tcp   filtered netbios-ssn  
445/tcp   filtered microsoft-ds  
MAC Address: 34:23:07:7A:4A:59 (Hon Hai Precision Ind.)  
  
Nmap done: 1 IP address (1 host up) scanned in 3.66 seconds  
root@kali:~#
```

So, see here is a difference in scanning in port 80 http service is running, now using TCP scan see what happened. So, for TCP scan we need to use the small s capital T option ok, the result would be same. Now, suppose using the UDP scan we used to scan the port 53. So, for UDP scan we need to use as small s capital U option and we are going to scan the port 53. So, p then 53 port number and then the IP address; no it cannot able to find out the port 53 in UDP scan. I am using UDP scan, I am going to scan another host that is the IP address 192.168.0.106 ok; port 53 UDP port is basically closed.

Now, I am performing ACK scan, for ACK scan we need to use the small s capital A and then suppose I am going to scan first 1000 port for the IP address 192.168.0.106 ok. See port 139 and port 445 tcp port is basically filtered ok. So now, I am going to use some output option. So, suppose I want to store this result in a txt file; so, for txt output we need to use that small o capital option followed by the filename. Suppose, here the file name is *scan.txt* and the location is root plus myfile ok. Now, go to the location, now this is route and that is my folders my file and then this is the result *scan.txt*.

(Refer Slide Time: 24:43)

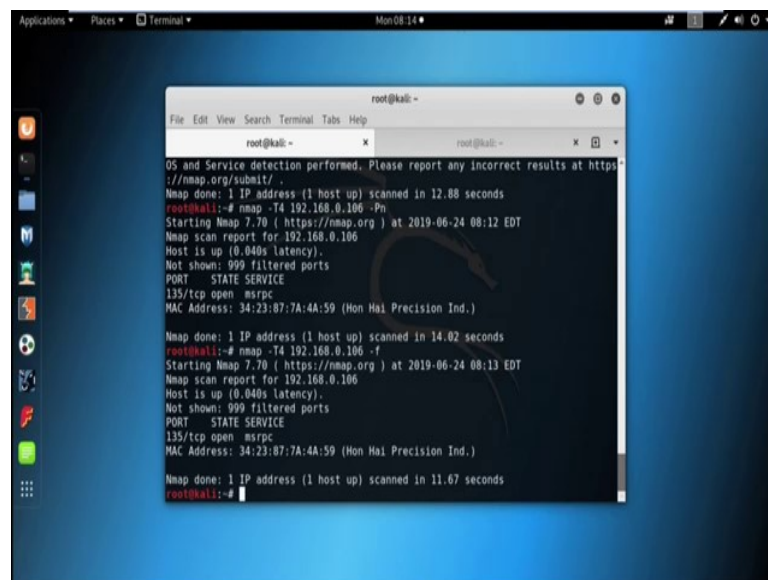


```
# Nmap 7.70 scan initiated Mon Jun 24 08:08:42 2019 as: nmap -sA -T4 -p 1-1000 -oN /root/myfile/scan.txt 192.168.0.106
Nmap scan report for 192.168.0.106
Host is up (0.017s latency).
Not shown: 998 unfiltered ports
PORT      STATE SERVICE
139/tcp    filtered netbios-ssn
445/tcp    filtered microsoft-ds
MAC Address: 34:23:07:7A:4A:59 (Hon Hai Precision Ind.)

# Nmap done at Mon Jun 24 08:08:45 2019 -- 1 IP address (1 host up) scanned in 3.66 seconds
```

See so, this way we can store the result in a text file and also in xml file and greppable option is also available.

(Refer Slide Time: 25:05)



```
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/.
Nmap done: 1 IP address (1 host up) scanned in 12.88 seconds
root@kali:~# nmap -T4 192.168.0.106 -Pn
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-24 08:12 EDT
Nmap scan report for 192.168.0.106
Host is up (0.040s latency).
Not shown: 999 filtered ports
PORT      STATE SERVICE
135/tcp    open  msrpc
MAC Address: 34:23:07:7A:4A:59 (Hon Hai Precision Ind.)

Nmap done: 1 IP address (1 host up) scanned in 14.02 seconds
root@kali:~# nmap -T4 192.168.0.106 -f
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-24 08:13 EDT
Nmap scan report for 192.168.0.106
Host is up (0.040s latency).
Not shown: 999 filtered ports
PORT      STATE SERVICE
135/tcp    open  msrpc
MAC Address: 34:23:07:7A:4A:59 (Hon Hai Precision Ind.)

Nmap done: 1 IP address (1 host up) scanned in 11.67 seconds
root@kali:~#
```

Now, we are, scan is also there to find out the operating system for a particular host. So, for operating system we need to use the option dash capital O ok; see that is the result Microsoft Windows ok. So, the victim machine basically run over Microsoft Windows. Now, there is also some other scanning option are also available like aggressive scan; *nmap* then suppose I am using the timing option T4, then for aggressive scan we can use

the option dash capital A and then the IP address of the effective machine. So, by doing the aggressive scan we can find out the service scan and operating system scan together.

It gives us the result about some common services which is running in particular port and as well as the operating system also. Now, see PORT 80 is open and TP LINK WR 740N WAP http configuration version are there and 1900 TCP port is open upnp service is running and the version is ipOS upnp and the thing is that operating system is linux with the kernel 2.6. So, we got all these details by using the aggressive scan, sometimes firewall also block the ping request sent by the tool and map. So, in that case we can use the option dash capital P small n to bypass the firewall; so, it basically no pings scan.

See that is now packet fragmentation, an *nmap* scan will use tiny IP fragments if that dash small f is specified, by default *nmap* will include up to 8 bytes of data in each fragment. So, a typical 20 or 24 byte TCP packet is sent in 3 tiny fragments, every instance of dash small f adds 8 to the maximum fragmented data size; see that is the result.