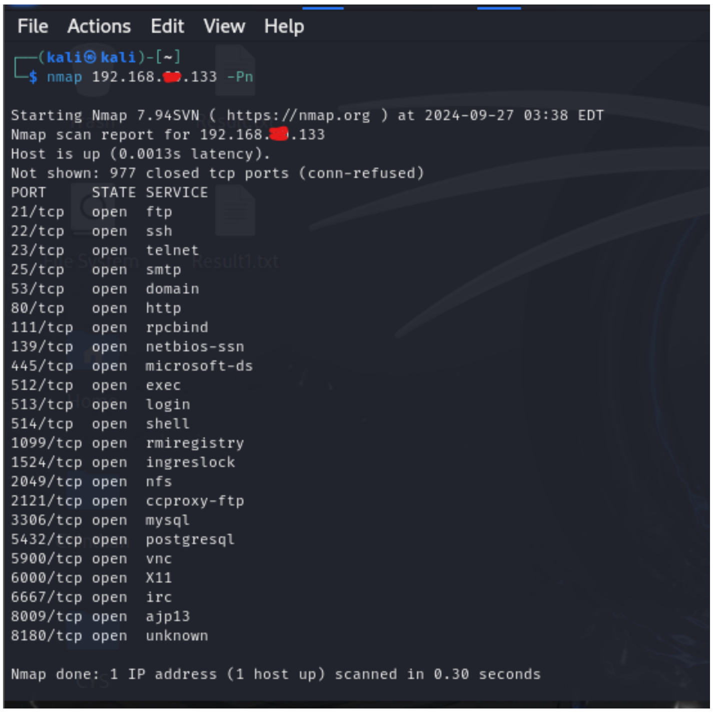
Network Scanning

NMAP:

- **nmap <ip-address> -Pn :**
  - **-Pn:** This flag tells Nmap to skip the ping check, assuming the host is up. Normally, Nmap tries to ping the target to see if it's online, but with -Pn, it skips that step and proceeds with the port scan directly.
  - This can be used for windows OS (Target machine) if host is up because normal nmap scanning (nmap <ip-address>) is blocked by the windows firewall.

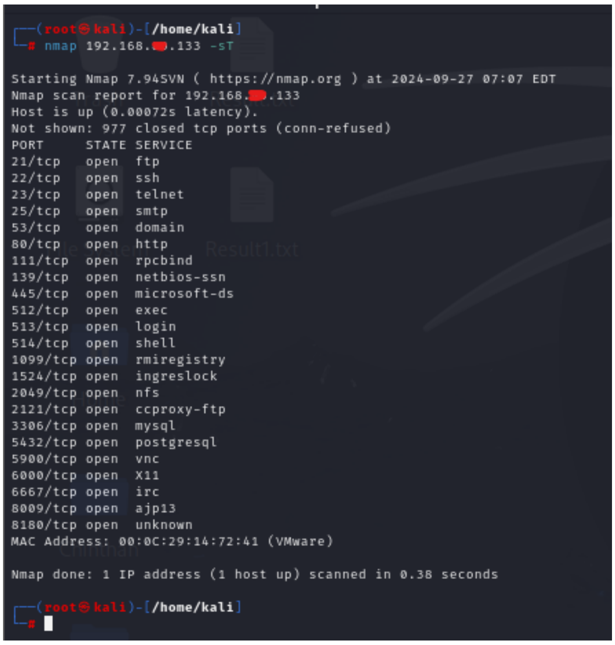


```
┌──(kali㉿kali)-[~]
└─$ nmap 192.168.[redacted].133 -Pn

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-27 03:38 EDT
Nmap scan report for 192.168.[redacted].133
Host is up (0.0013s latency).
Not shown: 977 closed tcp ports (conn-refused)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 0.30 seconds
```

Different Types of Scanning Techniques:

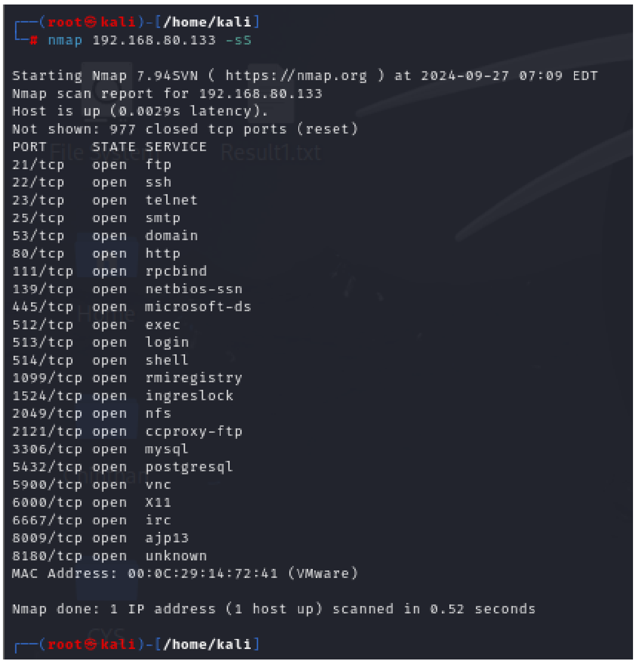
- **Tcp connect scan (-sT): (nmap <ip-address> -sT)**
  - A TCP Connect Scan (-sT) is one of the most basic types of scans in network security. It works by attempting to make a full TCP connection with the target system.
  - SYN: The scanner sends a TCP SYN packet to the target port, initiating the connection.
  - SYN-ACK: If the port is open, the target responds with a SYN-ACK packet, acknowledging the request.

- ACK: The scanner sends an ACK packet, completing the connection. After this, the scanner immediately sends a RST (reset) packet to close the connection and move on to scan other ports.



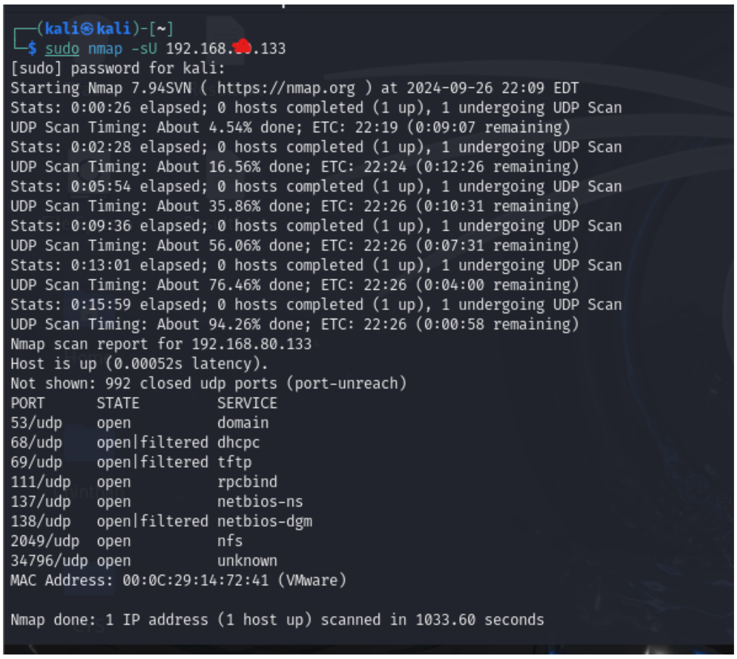
```
┌──(root㉿kali)-[/home/kali]
└─# nmap 192.168.[redacted].133 -sT

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-27 07:07 EDT
Nmap scan report for 192.168.[redacted].133
Host is up (0.00072s latency).
Not shown: 977 closed tcp ports (conn-refused)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 00:0C:29:14:72:41 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.38 seconds

┌──(root㉿kali)-[/home/kali]
└─# █
```

- **SYN scan (-sS): (nmap <ip-address> -sS)**
  - A **SYN Scan** (-sS) is a popular and efficient scanning technique used to check the status of TCP ports. It's often referred to as **half-open scanning** because it doesn't complete the full TCP handshake, making it stealthier than a TCP Connect Scan.
  - **SYN**: The scanner sends a TCP SYN packet to the target port, initiating a connection request.
  - **SYN-ACK**: If the port is open, the target responds with a SYN-ACK (synchronization-acknowledgment) packet.
  - **RST**: Instead of completing the handshake with an ACK packet, the scanner sends an RST (reset) packet to terminate the connection. This prevents a full connection from being established.



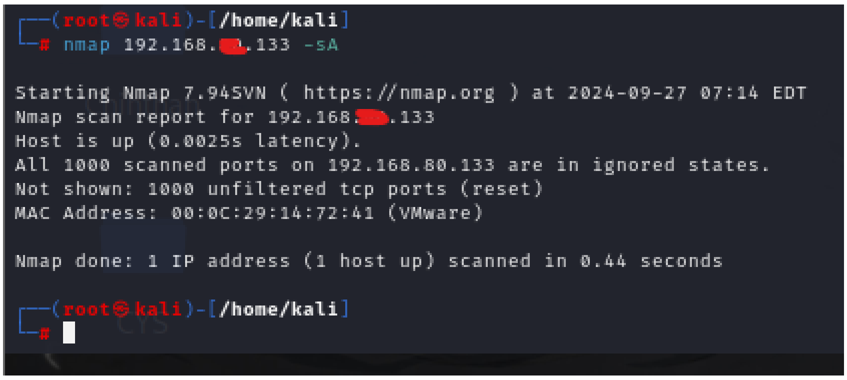
```
┌──(root㉿kali)-[/home/kali]
└─# nmap 192.168.80.133 -sS

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-27 07:09 EDT
Nmap scan report for 192.168.80.133
Host is up (0.0029s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 00:0C:29:14:72:41 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.52 seconds

┌──(root㉿kali)-[/home/kali]
```

- ### UDP scan (-sU): (nmap <ip-address> -sU)
  - Scans UDP (User Datagram Protocol) ports to detect open services like DNS, SNMP, and others that use UDP.
  - Unlike TCP, UDP is connectionless, making it harder to detect open ports reliably since there's no acknowledgment for open ports.
  - Challenges: Often slower because there's no guaranteed response for open ports. Nmap may mark a port as "open|filtered" if it gets no response or "closed" if it receives an ICMP "port unreachable" message.
  - Common Uses: Checking for services like DNS (53), SNMP (161), and DHCP (67, 68)



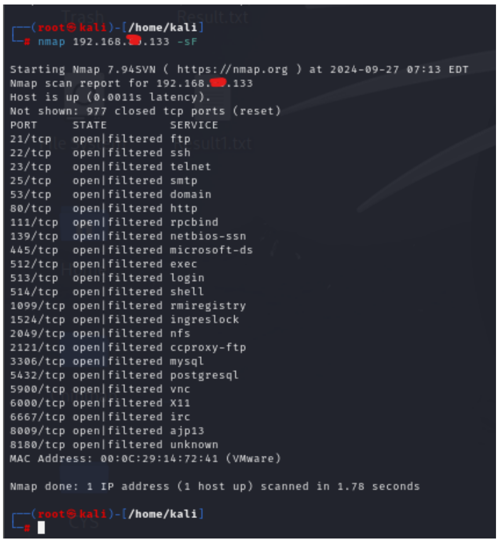
```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sU 192.168.[redacted].133
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-26 22:09 EDT
Stats: 0:00:26 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 4.54% done; ETC: 22:19 (0:09:07 remaining)
Stats: 0:02:28 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 16.56% done; ETC: 22:24 (0:12:26 remaining)
Stats: 0:05:54 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 35.86% done; ETC: 22:26 (0:10:31 remaining)
Stats: 0:09:36 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 56.06% done; ETC: 22:26 (0:07:31 remaining)
Stats: 0:13:01 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 76.46% done; ETC: 22:26 (0:04:00 remaining)
Stats: 0:15:59 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 94.26% done; ETC: 22:26 (0:00:58 remaining)
Nmap scan report for 192.168.80.133
Host is up (0.00052s latency).
Not shown: 992 closed udp ports (port-unreach)
PORT      STATE         SERVICE
53/udp    open          domain
68/udp    open|filtered dhcpc
69/udp    open|filtered tftp
111/udp   open          rpcbind
137/udp   open          netbios-ns
138/udp   open|filtered netbios-dgm
2049/udp  open          nfs
34796/udp open          unknown
MAC Address: 00:0C:29:14:72:41 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1033.60 seconds
```

- **ACK scan (-sA): (nmap <ip-address> -sA)**
  - A TCP ACK scan sends packets with the ACK flag set to determine if a port is filtered or unfiltered.
  - This scan does not detect open ports directly but is used to identify firewall rules and whether ports are filtered (blocked) or unfiltered (allowed).
  - Filtered: No response or an ICMP "destination unreachable" message.
  - Unfiltered: Receives an RST (Reset) response.
  - Common Uses: To check if a firewall allows traffic through certain ports but does not provide information on whether the port is actually open.
  - This scans are used for identifying firewall and IDS devices.



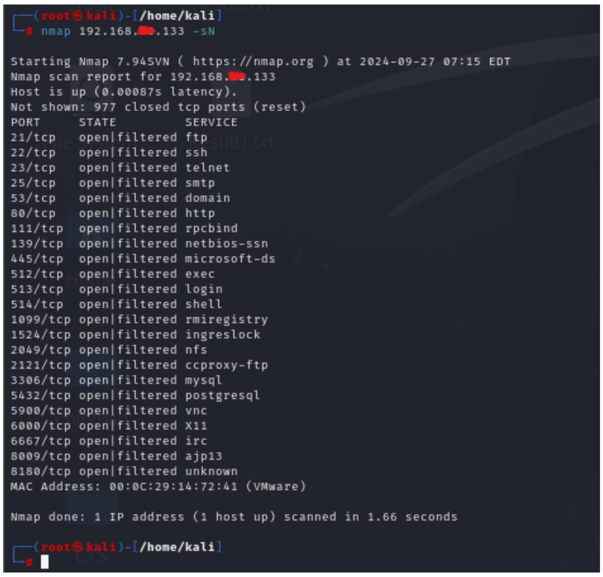
```
┌──(root㉿kali)-[/home/kali]
└─# nmap 192.168.[redacted].133 -sA

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-27 07:14 EDT
Nmap scan report for 192.168[redacted].133
Host is up (0.0025s latency).
All 1000 scanned ports on 192.168.80.133 are in ignored states.
Not shown: 1000 unfiltered tcp ports (reset)
MAC Address: 00:0C:29:14:72:41 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.44 seconds

┌──(root㉿kali)-[/home/kali]
└─# 
```

- **TCP FIN Scan (-sF): (nmap <ip-address> -sF)**
  - The FIN scan sends a packet with the **FIN** (Finish) flag set, which normally signals the end of a TCP connection.
  - Closed ports should respond with an **RST (Reset)** packet.
  - Open ports typically **ignore the packet**, so no response is received.
  - Stealthy because many firewalls and IDS/IPS systems do not log packets with only the FIN flag, making it less likely to trigger alarms.
  - Some systems do not respond to this scan in a predictable manner, and it might not work well on newer operating systems like Windows.



```
┌──(root㉿kali)-[/home/kali]
└─# nmap 192.168.[redacted].133 -sF

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-27 07:13 EDT
Nmap scan report for 192.168.[redacted].133
Host is up (0.0011s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE         SERVICE
21/tcp    open|filtered ftp
22/tcp    open|filtered ssh
23/tcp    open|filtered telnet
25/tcp    open|filtered smtp
53/tcp    open|filtered domain
80/tcp    open|filtered http
111/tcp   open|filtered rpcbind
139/tcp   open|filtered netbios-ssn
445/tcp   open|filtered microsoft-ds
512/tcp   open|filtered exec
513/tcp   open|filtered login
514/tcp   open|filtered shell
1099/tcp open|filtered rmiregistry
1524/tcp open|filtered ingreslock
2049/tcp open|filtered nfs
2121/tcp open|filtered ccproxy-ftp
3306/tcp open|filtered mysql
5432/tcp open|filtered postgresql
5900/tcp open|filtered vnc
6000/tcp open|filtered X11
6667/tcp open|filtered irc
8009/tcp open|filtered ajp13
8180/tcp open|filtered unknown
MAC Address: 00:0C:29:14:72:41 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.78 seconds

┌──(root㉿kali)-[/home/kali]
└─# 
```

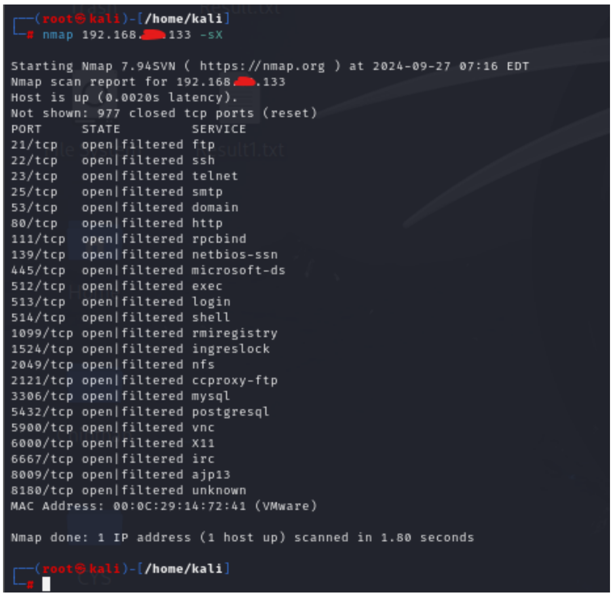
- **TCP NULL Scan (-sN): (nmap <ip-address> -sN)**
    - The NULL scan sends a completely **empty packet** with no flags set.
    - Closed ports will respond with an **RST**.
    - Open ports will **ignore** the packet.
    - Stealthy and can bypass some firewalls that only look for SYN or ACK flags.
    - Like the FIN scan, the NULL scan may not work effectively against certain systems (like Windows-based systems) because they might not respond in a standardized way.



```
┌──(root㉿kali)-[/home/kali]
└─# nmap 192.168.[redacted].133 -sN

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-27 07:15 EDT
Nmap scan report for 192.168.[redacted].133
Host is up (0.00087s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE         SERVICE
21/tcp    open|filtered ftp
22/tcp    open|filtered ssh
23/tcp    open|filtered telnet
25/tcp    open|filtered smtp
53/tcp    open|filtered domain
80/tcp    open|filtered http
111/tcp   open|filtered rpcbind
139/tcp   open|filtered netbios-ssn
445/tcp   open|filtered microsoft-ds
512/tcp   open|filtered exec
513/tcp   open|filtered login
514/tcp   open|filtered shell
1099/tcp open|filtered rmiregistry
1524/tcp open|filtered ingreslock
2049/tcp open|filtered nfs
2121/tcp open|filtered ccproxy-ftp
3306/tcp open|filtered mysql
5432/tcp open|filtered postgresql
5900/tcp open|filtered vnc
6000/tcp open|filtered X11
6667/tcp open|filtered irc
8009/tcp open|filtered ajp13
8180/tcp open|filtered unknown
MAC Address: 00:0C:29:14:72:41 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.66 seconds

┌──(root㉿kali)-[/home/kali]
└─# 
```

- **TCP XMAS Scan (-sX): (nmap <ip-address> -sX)**
    - The XMAS scan sends a packet with the FIN, PSH, and URG flags set, lighting up the TCP header like a Christmas tree, hence the name.
    - Closed ports respond with an RST.
    - Open ports ignore the packet, as they do in FIN and NULL scans.
    - Stealthy and can slip through poorly configured firewalls.

- Similar to the FIN and NULL scans, it may not work effectively on systems like Windows.



```
┌──(root㉿kali)-[/home/kali]
└─# nmap 192.168.[redacted].133 -sX

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-27 07:16 EDT
Nmap scan report for 192.168.[redacted].133
Host is up (0.0020s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE         SERVICE
21/tcp   open|filtered ftp
22/tcp   open|filtered ssh
23/tcp   open|filtered telnet
25/tcp   open|filtered smtp
53/tcp   open|filtered domain
80/tcp   open|filtered http
111/tcp  open|filtered rpcbind
139/tcp  open|filtered netbios-ssn
445/tcp  open|filtered microsoft-ds
512/tcp  open|filtered exec
513/tcp  open|filtered login
514/tcp  open|filtered shell
1099/tcp open|filtered rmiregistry
1524/tcp open|filtered ingreslock
2049/tcp open|filtered nfs
2121/tcp open|filtered ccproxy-ftp
3306/tcp open|filtered mysql
5432/tcp open|filtered postgresql
5900/tcp open|filtered vnc
6000/tcp open|filtered X11
6667/tcp open|filtered irc
8009/tcp open|filtered ajp13
8180/tcp open|filtered unknown
MAC Address: 00:0C:29:14:72:41 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.80 seconds

┌──(root㉿kali)-[/home/kali]
└─# 
```

All these requires Root Privileges.

WIRESHARK

- Wireshark is a popular network protocol analyzer used to capture, analyze, and troubleshoot network traffic.
- Wireshark can be used when network scanners (like Nmap) are blocked or defended by firewalls.
- Bypassing Scanner Limitations
- Packet Analysis of Allowed Traffic
- Firewall and IDS Behavior Detection
- Traffic Inspection Without Triggering Alerts
- Identifying Anomalies

Toolbars:

1. Main Toolbar

The main toolbar gives you easy access to commonly used features like capturing packets, stopping the capture, saving files, and opening captures.

- **Start Capture**: Begins packet capture.
- **Stop Capture**: Stops an ongoing packet capture.
- **Open**: Opens a previously saved capture file.
- **Save**: Saves the current capture.
- **Restart**: Restarts the capture process without having to stop and then start again.
- **Find**: Allows you to search for specific packets.
- **Preferences**: Opens the settings menu.
- **Help**: Opens Wireshark documentation.

2. Display Filter Toolbar

This is one of the most used toolbars in Wireshark, allowing you to create filters to view specific network traffic based on protocols, addresses, or packet content.

- **Filter Box**: Enter display filters here (e.g., ip.addr == 192.168.1.1).
- **Apply**: Applies the display filter.
- **Clear**: Clears the current filter.
- **Expression**: Opens a dialog to help build complex filters.

3. Packet List Toolbar

This toolbar is available when viewing captured packets and provides buttons to navigate through the packet list.

- **First Packet**: Jumps to the first packet.
- **Previous Packet**: Moves to the previous packet.

- **Next Packet**: Moves to the next packet.
- **Last Packet**: Jumps to the last packet.

4. Packet Details Toolbar

When selecting individual packets, this toolbar allows you to navigate and manipulate the detailed information of a packet.

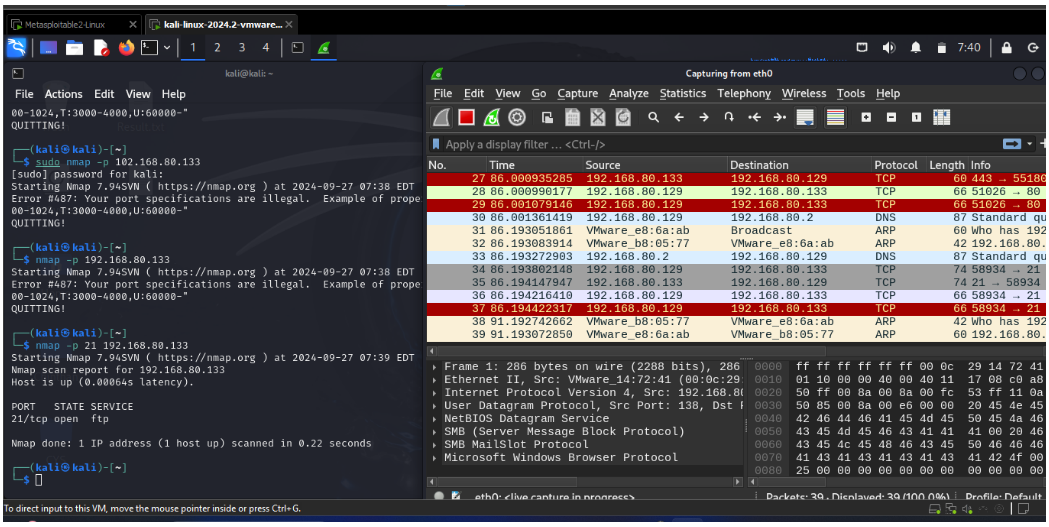
- **Expand All**: Expands all the layers of a packet.
- **Collapse All**: Collapses all the layers of a packet.

5. Status Bar

Though not technically a toolbar, the status bar at the bottom provides quick information about the capture such as the number of packets, elapsed capture time, and the applied filter.

You can customize the layout of toolbars by navigating to View > Layout in Wireshark.

- **nmap -p <port-number> <ip-address>**


Metasploitable2-Linux
kali-linux-2024.2-vmware...
1 2 3 4
7:40
kali@kali: ~
File Actions Edit View Help
Capturing from eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
00-1024,T:3000-4000,U:60000-"
QUITTING!

┌──(kali㉿kali)-[~]
└─$ sudo nmap -p 102.168.80.133
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-27 07:38 EDT
Error #487: Your port specifications are illegal.  Example of prope
00-1024,T:3000-4000,U:60000-"
QUITTING!

┌──(kali㉿kali)-[~]
└─$ nmap -p 192.168.80.133
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-27 07:38 EDT
Error #487: Your port specifications are illegal.  Example of prope
00-1024,T:3000-4000,U:60000-"
QUITTING!

┌──(kali㉿kali)-[~]
└─$ nmap -p 21 192.168.80.133
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-27 07:39 EDT
Nmap scan report for 192.168.80.133
Host is up (0.00064s latency).

PORT   STATE SERVICE
21/tcp open  ftp

Nmap done: 1 IP address (1 host up) scanned in 0.22 seconds

┌──(kali㉿kali)-[~]
└─$ 
```

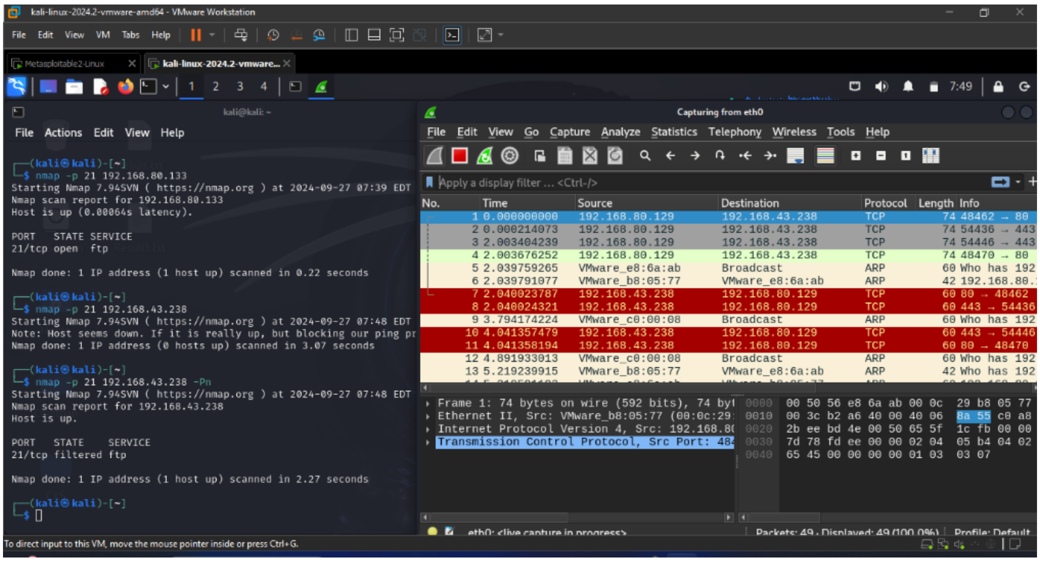
No.	Time	Source	Destination	Protocol	Length	Info
27	86.000935285	192.168.80.133	192.168.80.129	TCP	60	443 → 55180
28	86.000990177	192.168.80.129	192.168.80.133	TCP	66	51026 → 80
29	86.001079146	192.168.80.129	192.168.80.133	TCP	66	51026 → 80
30	86.001361419	192.168.80.129	192.168.80.2	DNS	87	Standard qu
31	86.193051861	VMware_e8:6a:ab	Broadcast	ARP	60	Who has 192
32	86.193083914	VMware_b8:05:77	VMware_e8:6a:ab	ARP	42	192.168.80.
33	86.193272903	192.168.80.2	192.168.80.129	DNS	87	Standard qu
34	86.193802148	192.168.80.129	192.168.80.133	TCP	74	58934 → 21
35	86.194147947	192.168.80.133	192.168.80.129	TCP	74	21 → 58934
36	86.194216410	192.168.80.129	192.168.80.133	TCP	66	58934 → 21
37	86.194422317	192.168.80.129	192.168.80.133	TCP	66	58934 → 21
38	91.192742662	VMware_b8:05:77	VMware_e8:6a:ab	ARP	42	Who has 192
39	91.193072850	VMware_e8:6a:ab	VMware_b8:05:77	ARP	60	192.168.80.

```
▸ Frame 1: 286 bytes on wire (2288 bits), 286    0000  ff ff ff ff ff ff 00 0c  29 14 72 41
▸ Ethernet II, Src: VMware_14:72:41 (00:0c:29    0010  01 10 00 00 40 00 40 11  17 08 c0 a8
▸ Internet Protocol Version 4, Src: 192.168.8(   0020  50 ff 00 8a 00 8a 00 fc  53 ff 11 0a
▸ User Datagram Protocol, Src Port: 138, Dst P   0030  50 85 00 8a 00 e6 00 00  20 45 4e 45
▸ NetBIOS Datagram Service                       0040  42 46 44 46 41 45 4d 45  50 45 4a 46
▸ SMB (Server Message Block Protocol)            0050  43 45 4d 45 46 43 41 41  41 00 20 46
▸ SMB MailSlot Protocol                          0060  43 45 4c 45 48 46 43 45  50 46 46 46
▸ Microsoft Windows Browser Protocol             0070  41 43 41 43 41 43 41 43  41 42 4f 00
                                                 0080  25 00 00 00 00 00 00 00  00 00 00 00
```

eth0: <live capture in progress> | Packets: 39 · Displayed: 39 (100.0%) | Profile: Default

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

  ○ When command is typed on the terminal, it captured the packet of TCP handshake of the respective port.



kali-linux-2024.2-vmware-amd64 - VMware Workstation
File Edit View VM Tabs Help
Metasploitable2-Linux
kali-linux-2024.2-vmware...
7:49
kali@kali: ~
File Actions Edit View Help
Capturing from eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

```
┌──(kali㉿kali)-[~]
└─$ nmap -p 21 192.168.80.133
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-27 07:39 EDT
Nmap scan report for 192.168.80.133
Host is up (0.00064s latency).

PORT   STATE SERVICE
21/tcp open  ftp

Nmap done: 1 IP address (1 host up) scanned in 0.22 seconds

┌──(kali㉿kali)-[~]
└─$ nmap -p 21 192.168.43.238
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-27 07:48 EDT
Note: Host seems down. If it is really up, but blocking our ping pr
Nmap done: 1 IP address (0 hosts up) scanned in 3.07 seconds

┌──(kali㉿kali)-[~]
└─$ nmap -p 21 192.168.43.238 -Pn
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-27 07:48 EDT
Nmap scan report for 192.168.43.238
Host is up.

PORT   STATE    SERVICE
21/tcp filtered ftp

Nmap done: 1 IP address (1 host up) scanned in 2.27 seconds

┌──(kali㉿kali)-[~]
└─$ ▯
```

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.80.129	192.168.43.238	TCP	74	48462 → 80
2	0.000214073	192.168.80.129	192.168.43.238	TCP	74	54436 → 443
3	2.003404239	192.168.80.129	192.168.43.238	TCP	74	54446 → 443
4	2.003676252	192.168.80.129	192.168.43.238	TCP	74	48470 → 80
5	2.039759265	VMware_e8:6a:ab	Broadcast	ARP	60	Who has 192
6	2.039791077	VMware_b8:05:77	VMware_e8:6a:ab	ARP	42	192.168.80.
7	2.040023787	192.168.43.238	192.168.80.129	TCP	60	80 → 48462
8	2.040024321	192.168.43.238	192.168.80.129	TCP	60	443 → 54436
9	3.794174224	VMware_c0:00:08	Broadcast	ARP	60	Who has 192
10	4.041357479	192.168.43.238	192.168.80.129	TCP	60	443 → 54446
11	4.041358194	192.168.43.238	192.168.80.129	TCP	60	80 → 48470
12	4.891933013	VMware_c0:00:08	Broadcast	ARP	60	Who has 192
13	5.219239915	VMware_b8:05:77	VMware_e8:6a:ab	ARP	42	Who has 192

```
▸ Frame 1: 74 bytes on wire (592 bits), 74 byt    0000  00 50 56 e8 6a ab 00 0c  29 b8 05 77
▸ Ethernet II, Src: VMware_b8:05:77 (00:0c:29:    0010  00 3c b2 a6 40 00 40 06  8a 55 c0 a8
▸ Internet Protocol Version 4, Src: 192.168.80    0020  2b ee bd 4e 00 50 65 5f  1c fb 00 00
▸ Transmission Control Protocol, Src Port: 484    0030  7d 78 fd ee 00 00 02 04  05 b4 04 02
                                                  0040  65 45 00 00 00 00 01 03  03 07
```

eth0: <live capture in progress> Packets: 49 · Displayed: 49 (100.0%) Profile: Default

- The above command worked on windows OS.

THANK YOU

Wireshark Functions/operations are continued in Next Documentation Part