ASSESSMENT REPORT

- **Name:** Jahnavi
- **Target Domain:** scanme.nmap.org
- **Target IP:** (The Windows Host IP address used for Phases 2 & 3)
- **Summary**
  - ➢ Intially targets the website:scanme.nmap.org, using public tools(iplookup,whois,dnslookup) to know the details(ip addr, owner, geo-location, dns records)
  - ➢ Using nmap, performed network scanning against windows host, gathering the details of OS, version, banners, state of ports, services on the ports
  - ➢ Using bettercap targeting the windows host network activities , activities happenting on the same LAN

Phase 1: Footprinting & OSINT

The focus here is on **information gathering** and **documentation**.

Task Requirement	**Evidence / Findings**	**Command / Tool Used**
1. Determine IP and Location	IP Address: 45.33.32.156	WHOIS/IP Lookup
	Location (City/Country): California/Fremont/united States	
2. DNS Enumeration	Name Servers (NS Records): ns1.linode.com. ns4.linode.com. ns2.linode.com. ns3.linode.com. ns5.linode.com.	DNS Lookup
	Mail Exchange (MX Records):	

Task Requirement	Evidence / Findings	Command / Tool Used
	ASPMX.L.GOOGLE.COM. ALT1.ASPMX.L.GOOGLE.COM. ALT2.ASPMX.L.GOOGLE.COM. ASPMX3.GOOGLEMAIL.COM. ASPMX2.GOOGLEMAIL.COM.	
3. Advanced Search (Google Dork)	The Dork Used: **filetype:pdf site:nmap.org**	(N/A)
	Description of Document Found: nmap-mindmap.pdf – displays the commands used in nmap discovery.pdf – gives detailed explanation about discovering the host	

Challenge Questions (Phase 1)

Question 1 (Footprinting): According to the public records, who is the **Administrative Contact** for the nmap.org domain?

Answer: DYNADOT LLC Dynadot Inc

IANA ID: 472

URL: http://www.dynadot.com

Whois Server: whois.dynadot.com

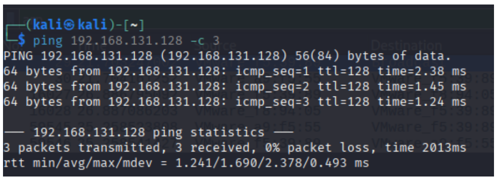
Question 2 (Ethical Hacking): Why is running a basic **Traceroute** command considered a more passive reconnaissance technique than running an **Nmap SYN scan**?

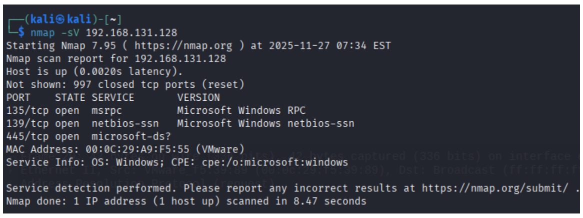
Answer: tracert/traceroute command gives the information about number of hops, TTL(time to live), size of the packets between source & destination where as **nmap**

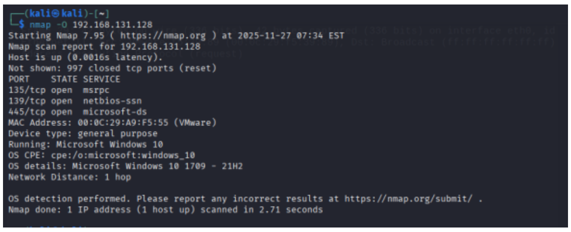
syn scan synchronises with the target by giving information like number of live host, open ports, banner info etc

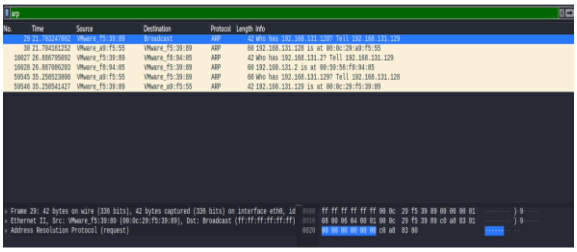
Phase 2: Network Scanning and Enumeration

The focus here is on **command execution** and **output analysis**.

<table>
<tr><th>Task Requirement</th><th>Command Used / Output</th><th>Proof of Execution</th></tr>
<tr><td><b>1. Host IP Identification</b></td><td>Output of <b>ping 192.168.131.128 -c 3</b></td><td>
<pre>
┌──(kali㉿kali)-[~]
└─$ ping 192.168.131.128 -c 3
PING 192.168.131.128 (192.168.131.128) 56(84) bytes of data.
64 bytes from 192.168.131.128: icmp_seq=1 ttl=128 time=2.38 ms
64 bytes from 192.168.131.128: icmp_seq=2 ttl=128 time=1.45 ms
64 bytes from 192.168.131.128: icmp_seq=3 ttl=128 time=1.24 ms

─── 192.168.131.128 ping statistics ───
3 packets transmitted, 3 received, 0% packet loss, time 2013ms
rtt min/avg/max/mdev = 1.241/1.690/2.378/0.493 ms
</pre></td></tr>
<tr><td><b>2. Service Version Scan</b></td><td>Full Nmap Command Used:
<b>nmap -sV 192.168.131.128</b></td><td>
<pre>
┌──(kali㉿kali)-[~]
└─$ nmap -sV 192.168.131.128
Starting Nmap 7.95 (https://nmap.org) at 2025-11-27 07:34 EST
Nmap scan report for 192.168.131.128
Host is up (0.0020s latency).
Not shown: 997 closed tcp ports (reset)
PORT   STATE SERVICE   VERSION
135/tcp open msrpc   Microsoft Windows RPC
139/tcp open netbios-ssn   Microsoft Windows netbios-ssn
445/tcp open microsoft-ds?
MAC Address: 00:0C:29:A9:F5:55 (VMware)
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 8.47 seconds
</pre></td></tr>
<tr><td><b>3. OS Detection</b></td><td>Full Nmap Command Used:
<b>nmap -O 192.168.131.128</b></td><td>
<pre>
┌──(kali㉿kali)-[~]
└─$ nmap -O 192.168.131.128
Starting Nmap 7.95 (https://nmap.org) at 2025-11-27 07:34 EST
Nmap scan report for 192.168.131.128
Host is up (0.0016s latency).
Not shown: 997 closed tcp ports (reset)
PORT   STATE SERVICE
135/tcp open msrpc
139/tcp open netbios-ssn
445/tcp open microsoft-ds
MAC Address: 00:0C:29:A9:F5:55 (VMware)
Device type: general purpose
Running: Microsoft Windows 10
OS CPE: cpe:/o:microsoft:windows_10
OS details: Microsoft Windows 10 1709 - 21H2
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2.71 seconds
</pre></td></tr>
<tr><td><b>4. Traffic Analysis</b></td><td>The specific <b>ARP requests</b> identified in Wireshark.</td><td>
<pre>
arp
No.   Time   Source   Destination   Protocol Length Info
   29 21.703247002 VMware_f5:39:89 Broadcast   ARP   42 Who has 192.168.131.128? Tell 192.168.131.129
   30 21.704161252 VMware_a9:f5:55 VMware_f5:39:89 ARP   60 192.168.131.128 is at 00:0c:29:a9:f5:55
16027 26.886795092 VMware_f5:39:89 VMware_f8:94:05 ARP   42 Who has 192.168.131.2? Tell 192.168.131.129
16028 26.887086203 VMware_f8:94:05 VMware_f5:39:89 ARP   60 192.168.131.2 is at 00:50:56:f8:94:05
59545 35.258523808 VMware_a9:f5:55 VMware_f5:39:89 ARP   60 Who has 192.168.131.129? Tell 192.168.131.128
59546 35.258541427 VMware_f5:39:89 VMware_a9:f5:55 ARP   42 192.168.131.129 is at 00:0c:29:f5:39:89

› Frame 29: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface eth0, id   0000 ff ff ff ff ff ff 00 0c 29 f5 39 89 08 06 00 01   ········)·9·····
› Ethernet II, Src: VMware_f5:39:89 (00:0c:29:f5:39:89), Dst: Broadcast (ff:ff:ff:ff:ff:ff) 0010 08 00 06 04 00 01 00 0c 29 f5 39 89 c0 a8 83 81   ········)·9·····
› Address Resolution Protocol (request)   0020 00 00 00 00 00 00 c0 a8 83 80   ······ ··
</pre></td></tr>
</table>

Challenge Questions (Phase 2)

- **Q1 Identified Service/Port:**

  Service Name/Port: **ms-wbt-server/3389**

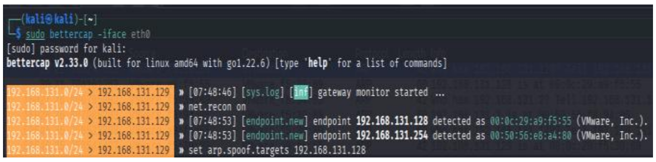
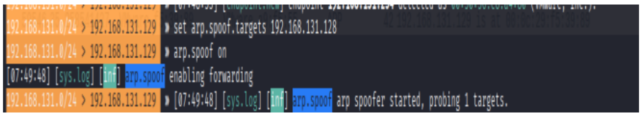
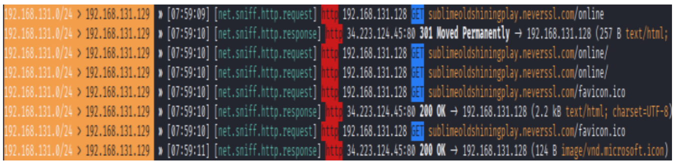
- **Q2** : Wireshark Display Filter: **arp**

- **Q3 Explanation of RST packets**:
  - If there is a connection to be established, source send a synchronisation to destination, when destination doesn't want to establish a connection it sends the RST flag

---

Phase 3: Sniffing and Traffic Analysis

The focus here is on **module execution** and **proof of intercept**.

Task Requirement	**Bettercap Command(s) Used**	**Proof of Execution**
1. Bettercap Initialization	Command to start Bettercap and run net.recon: **Sudo bettercap iface eth0** **net.recon on**	 ┌──(kali㉿kali)-[~] └─$ sudo bettercap -iface eth0 [sudo] password for kali: **bettercap v2.33.0** (built for linux amd64 with go1.22.6) [type '**help**' for a list of commands] 192.168.131.0/24 > 192.168.131.129 » [07:48:46] [sys.log] [inf] gateway monitor started ... 192.168.131.0/24 > 192.168.131.129 » net.recon on 192.168.131.0/24 > 192.168.131.129 » [07:48:53] [endpoint.new] endpoint **192.168.131.128** detected as 00:0c:29:a9:f5:55 (VMware, Inc.). 192.168.131.0/24 > 192.168.131.129 » [07:48:53] [endpoint.new] endpoint **192.168.131.254** detected as 00:50:56:e8:a4:80 (VMware, Inc.). 192.168.131.0/24 > 192.168.131.129 » set arp.spoof.targets 192.168.131.128
2. Targeted ARP Spoofing	set arp.spoof.targets 192.168.131.128 arp.spoof on	 192.168.131.0/24 > 192.168.131.129 » set arp.spoof.targets 192.168.131.128 192.168.131.0/24 > 192.168.131.129 » arp.spoof on [07:49:48] [sys.log] [inf] arp.spoof enabling forwarding 192.168.131.0/24 > 192.168.131.129 » [07:49:48] [sys.log] [inf] arp.spoof arp spoofer started, probing 1 targets.
3. Sniffing Setup	net.sniff on	(N/A)
4. Capture the Traffic	**Cleartext HTTP log line** from Bettercap confirming the visit to http://neverssl.com.	 192.168.131.0/24 > 192.168.131.129 » [07:59:09] [net.sniff.http.request] http 192.168.131.128 GET sublimeoldshiningplay.neverssl.com/online 192.168.131.0/24 > 192.168.131.129 » [07:59:10] [net.sniff.http.response] http 34.223.124.45:80 **301 Moved Permanently** → 192.168.131.128 (257 B text/html; 192.168.131.0/24 > 192.168.131.129 » [07:59:10] [net.sniff.http.request] http 192.168.131.128 GET sublimeoldshiningplay.neverssl.com/online/ 192.168.131.0/24 > 192.168.131.129 » [07:59:10] [net.sniff.http.request] http 192.168.131.128 GET sublimeoldshiningplay.neverssl.com/online/ 192.168.131.0/24 > 192.168.131.129 » [07:59:10] [net.sniff.http.request] http 192.168.131.128 GET sublimeoldshiningplay.neverssl.com/favicon.ico 192.168.131.0/24 > 192.168.131.129 » [07:59:10] [net.sniff.http.response] http 34.223.124.45:80 **200 OK** → 192.168.131.128 (2.2 kB text/html; charset=UTF-8) 192.168.131.0/24 > 192.168.131.129 » [07:59:10] [net.sniff.http.request] http 192.168.131.128 GET sublimeoldshiningplay.neverssl.com/favicon.ico 192.168.131.0/24 > 192.168.131.129 » [07:59:11] [net.sniff.http.response] http 34.223.124.45:80 **200 OK** → 192.168.131.128 (124 B image/vnd.microsoft.icon)

Challenge Questions (Phase 3)

- **Q1 Answer:** The three Bettercap commands used:

  sudo bettercap iface eth0

  set arp.spoof.targets 192.168.131.128

  arp.spoof on

```
net.sniff on
```

- **Q2** : Explanation of ARP Spoofing:

  Spoofs the mac address to make the target to connect with the machine thinking it is in same LANby following the process(broadcast- arp reply, arp response)

- **Q3 Countermeasure protocol and port:**

- https/443
- also can set security settings in the browser want to be allowed/block/ask(default)