Task 4

Process for analyzing the wireshark capture is using the different tabs in wireshark such as:

Endpoints: To find all ip addresses of hosts within this capture
IO graph: To visualize network traffic flow within this capture
Protocol hierarchy: To summarize the protocols used within this capture
Conversations: To summarize which hosts communicate the most and with which others
Filtering by protocol in the main display window allowing each protocol to be viewed separately to check for protocol-specific attacks

1) Traffic Overview

• Provide a statistical breakdown of the traffic. Analyze the total number of packets, protocols used, traffic distribution by protocol, and any other relevant statistics

Total number of packets : 62895
Protocols used:
UDP on IPv6 : 12
UDP on IPv4 : 2625
TCP on IPv4 : 56421
ICMP on IPv4 : 199
ARP : 3638

• List the top 5 IP addresses by traffic volume and speculate on their roles within the network

Top 5 ip addresses IPv4 :

```
[
  5.{
      "Address": "192.168.0.100",
      "Bytes": "269433",
       "Packets": "4257",
      "Rx Bytes": "124016",
      "Rx Packets": "2084",
      "Tx Bytes": "145417",
      "Tx Packets": "2173"
   },
   4.{
      "Address": "172.217.0.234",
      "Bytes": "7439392",
```

"Packets": "6028",
"Rx Bytes": "160502",
"Rx Packets": "2396",
"Tx Bytes": "7278890",
"Tx Packets": "3632"
},
3.{ The gateway router

"Address": "192.168.0.1",
"Bytes": "3570835",
"Packets": "11986",
"Rx Bytes": "473173",
"Rx Packets": "5881",
"Tx Bytes": "3097662",
"Tx Packets": "6105"
},

4.{ The victim
"Address": "192.168.0.105",
"Bytes": "18204895",
"Packets": "27187",
"Rx Bytes": "17098833",
"Rx Packets": "14431",
"Tx Bytes": "1106062",
"Tx Packets": "12756"
},
5.{ The attacker
"Address": "192.168.0.101",
"Bytes": "12530140",
"Packets": "35818",
"Rx Bytes": "10716662",
"Rx Packets": "16849",
"Tx Bytes": "1813478",
"Tx Packets": "18969"
}
]

Top addresses IPv6:

[
2.{
"Address": "ff02::fb",
"Bytes": "1224",
"Packets": "12",

"Rx Bytes": "1224",
"Rx Packets": "12",
"Tx Bytes": "0",
"Tx Packets": "0"
},
1.{
"Address": "fe80::b8bc:b3b6:de93:cad4",
"Bytes": "1224",
"Packets": "12",
"Rx Bytes": "0",
"Rx Packets": "0",
"Tx Bytes": "1224",
"Tx Packets": "12"
}
]

Top addresses Ethernet:

[
5.{
"Address": "08:00:27:98:93:c7",
"Bytes": "125 kB",
"Packets": "2,066",
"Rx Bytes": "124 kB",
"Rx Packets": "2,060",
"Tx Bytes": "360 bytes",
"Tx Packets": "6"
},
4.{
"Address": "70:20:84:08:71:cf",
"Bytes": "271 kB",
"Packets": "4,283",
"Rx Bytes": "124 kB",
"Rx Packets": "2,090",
"Tx Bytes": "147 kB",
"Tx Packets": "2,193"
},
3.{ the victim 192.168.0.105
"Address": "08:00:27:4f:41:8d",
"Bytes": "9 MB",
"Packets": "16,245",
"Rx Bytes": "9 MB",
"Rx Packets": "8,792",
"Tx Bytes": "620 kB",

"Tx Packets": "7,453"
},
2.{ the gateway router 2.168.0.1
"Address": "e4:6f:13:68:45:98",
"Bytes": "21 MB",
"Packets": "37,848",
"Rx Bytes": "2 MB",
"Rx Packets": "18,588",
"Tx Bytes": "19 MB",
"Tx Packets": "19,260"
},
1.{ The attacker 192.168.0.101
"Address": "08:00:27:9b:b5:04",
"Bytes": "31 MB",
"Packets": "62,418",
"Rx Bytes": "20 MB",
"Rx Packets": "28,435",
"Tx Bytes": "11 MB",
"Tx Packets": "33,983"
}
]

2) Protocol Analysis

Wireshark · Protocol Hierarchy Statistics · attack_trace_2024.pcapng

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s	PDUs
Frame	100.0	62895	100.0	30778949	107 k	0	0	0	62895
Ethernet	100.0	62895	3.0	921504	3,206	0	0	0	62895
Internet Protocol Version 6	0.0	12	0.0	480	1	0	0	0	12
User Datagram Protocol	0.0	12	0.0	96	0	0	0	0	12
Multicast Domain Name System	0.0	12	0.0	480	1	12	480	1	12
Internet Protocol Version 4	94.2	59245	3.8	1184900	4,123	0	0	0	59245
User Datagram Protocol	4.2	2625	0.1	21000	73	0	0	0	2625
Simple Service Discovery Protocol	0.7	443	0.4	125897	438	443	125897	438	443
Network Time Protocol	0.0	6	0.0	288	1	6	288	1	6
Multicast Domain Name System	0.0	12	0.0	480	1	12	480	1	12
Domain Name System	3.4	2142	1.1	339807	1,182	2142	339807	1,182	2142
Data	0.0	22	0.0	6600	22	22	6600	22	22
Transmission Control Protocol	89.7	56421	91.1	28036290	97 k	47791	18153972	63 k	56421
Transport Layer Security	11.9	7485	47.0	14477769	50 k	7485	11539175	40 k	8556
Hypertext Transfer Protocol	1.8	1144	14.8	4549501	15 k	583	236891	824	1144
Portable Network Graphics	0.0	4	0.0	2930	10	4	2930	10	4
Online Certificate Status Protocol	0.3	183	0.2	69390	241	183	69390	241	183
Media Type	0.1	63	9.3	2850216	9,918	63	2850216	9,918	63
Line-based text data	0.3	201	17.2	5294301	18 k	201	5294301	18 k	201
JPEG File Interchange Format	0.1	39	1.5	460206	1,601	39	460206	1,601	39
JavaScript Object Notation	0.0	3	0.0	687	2	2	552	1	3
HTML Form URL Encoded	0.0	4	0.0	430	1	4	430	1	4
eXtensible Markup Language	0.0	7	0.1	27391	95	7	27391	95	7
Compuserve GIF	0.1	58	0.1	28033	97	58	28033	97	58
Data	0.0	1	0.0	4	0	1	4	0	1
Internet Control Message Protocol	0.3	199	0.1	39263	136	182	34115	118	199
Domain Name System	0.0	3	0.0	336	1	3	336	1	3
Data	0.0	14	0.0	4200	14	14	4200	14	14
Address Resolution Protocol	5.8	3638	0.3	103844	361	3638	103844	361	3638

• Identify the different protocols present in the trace and the purpose they serve in the traffic

- **Ethernet:** The dominant LAN protocol responsible for moving data across wired networks.
- **Internet Protocol (IP) versions 4 (IPv4) and 6 (IPv6):** The workhorses of the internet layer, routing data packets across networks.
- **User Datagram Protocol (UDP):** A connectionless protocol for sending data datagrams with minimal overhead. UDP is commonly used for DNS requests and other time-sensitive applications where ordered delivery is not critical.
- **Transmission Control Protocol (TCP):** A connection-oriented protocol that ensures reliable delivery of data streams. TCP is used for applications like file transfer, web browsing, and email.
- **Hypertext Transfer Protocol (HTTPS):** The secure version of HTTP, used to encrypt communication between web servers and browsers.
- **Domain Name System (DNS):** A hierarchical and distributed naming system for computers, services, and other resources connected to the internet. DNS translates human-readable domain names into machine-readable IP addresses.
- **Simple Service Discovery Protocol (SSDP):** A discovery protocol used to advertise devices and services on a local network.
- **Network Time Protocol (NTP):** A protocol for synchronizing the clocks of computers over a network.
- **Media Type (MT):** A framework for identifying file formats and MIME types

• Look for any non-standard protocols or unusual usage of standard protocols

The protocols used are all standard, but some of the usage patterns (like image files directly under TCP) might be non-standard in the context of typical application behaviors. This could be characteristic of custom applications, direct file transmissions, or specific services not encapsulated by standard web traffic protocols.

Non-Standard or Unusual Aspects:

- JPEG File Interchange Format (JFIF), JavaScript Object Notation (JSON), and HTML Form URL Encoded data appearing directly under TCP instead of being encapsulated within HTTP or other higher-layer protocols could be unusual. However, these are more about the payload types than non-standard protocols.
- Portable Network Graphics (PNG) directly under TCP suggests direct transmission of PNG files over TCP without an enclosing protocol like HTTP. This is atypical but not necessarily non-standard.
- Compuserve GIF data is listed under TCP, which is uncommon as GIFs are typically transferred via HTTP in web contexts.

3) IP Address and Connection Mapping

• Map out the connections between different IP addresses with timeline (e.g. which IP connects to which first)

• Identify any external IP addresses and their interactions with internal IPs

There are many external IP addresses interacting with internal IP addresses
Only 192.168.0.101 and 192.168.0.105 received packets from external IP addresses
And only 192.168.0.1, 192.168.0.100, 192.168.0.101 and 192.168.0.105 sent to external IP addresses

4)Identification of Malicious Activities

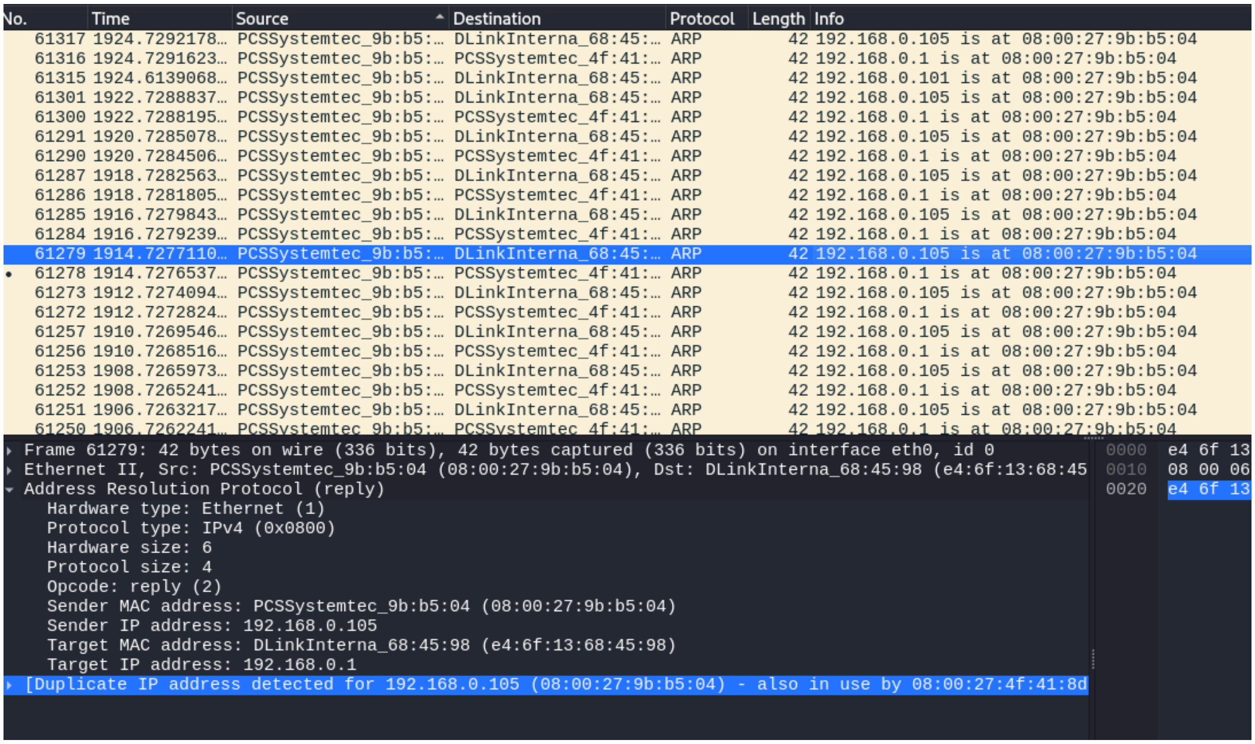
• Based on your analysis, identify any potential malicious actors.

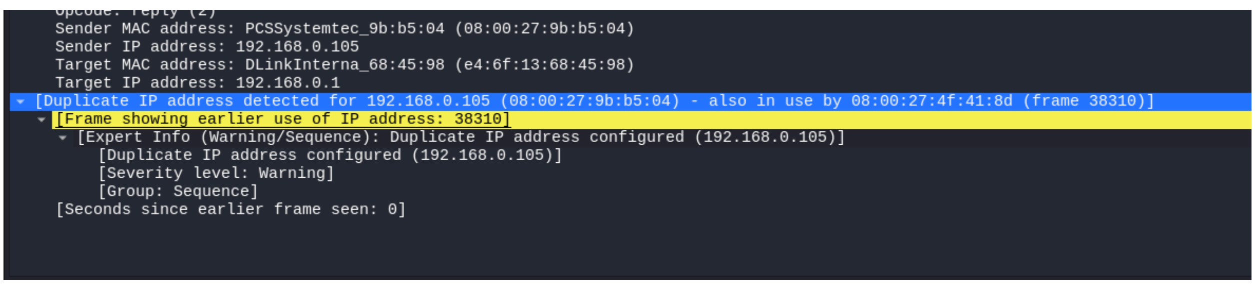
• Describe the vulnerabilities that have been exploited to carry out the attacks.

• Outline the sequence of successful attacks and what the attacker achieved after each phase

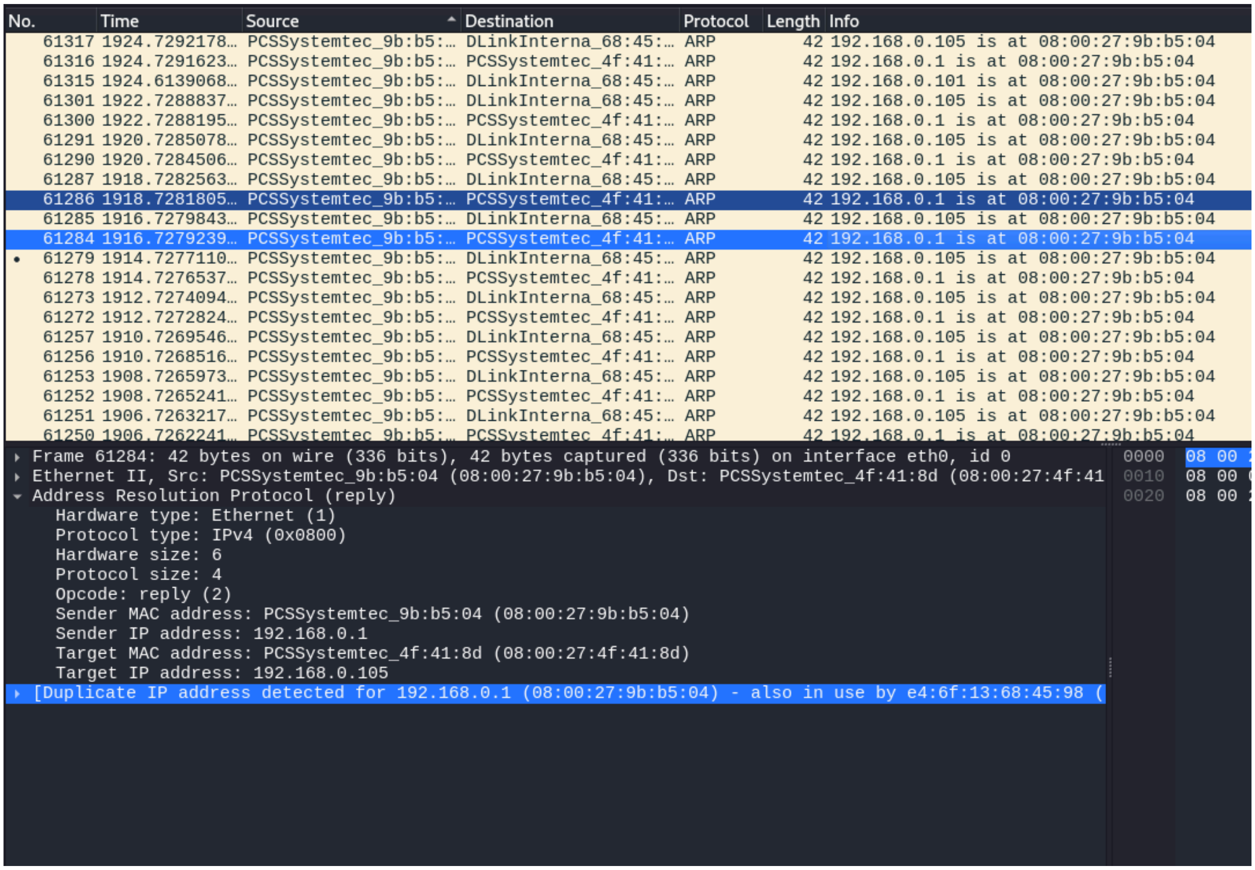
The IP address 192.168.0.101 is a malicious actor who performed ARP poisoning, ARP flooding attacks and TCP scanning

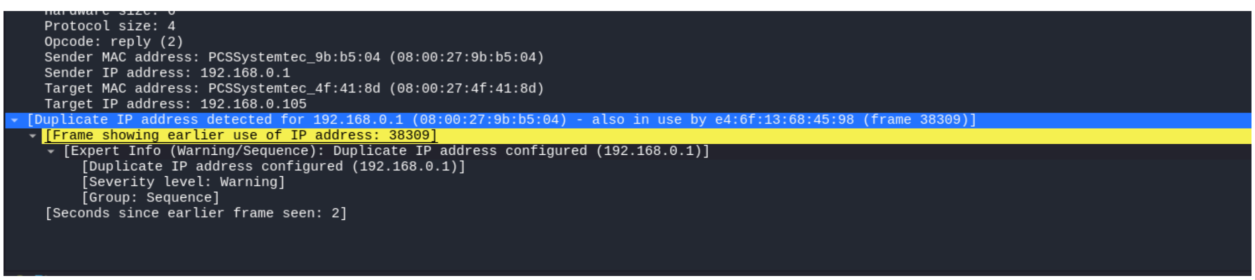
1. ARP poisoning





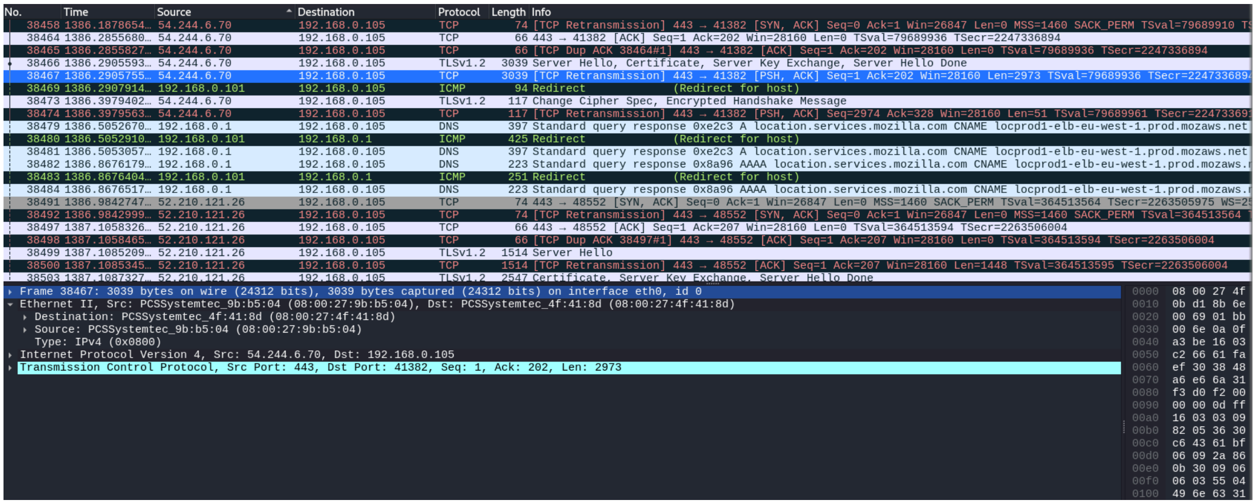
From the two screenshots we can see that 192.168.0.101 tries to make 192.168.0.1 map 192.168.0.105 to his mac address 08:00:27:9b:b5:04 instead of the right one 08:00:27:4f:41:8d by sending falsified arp packet



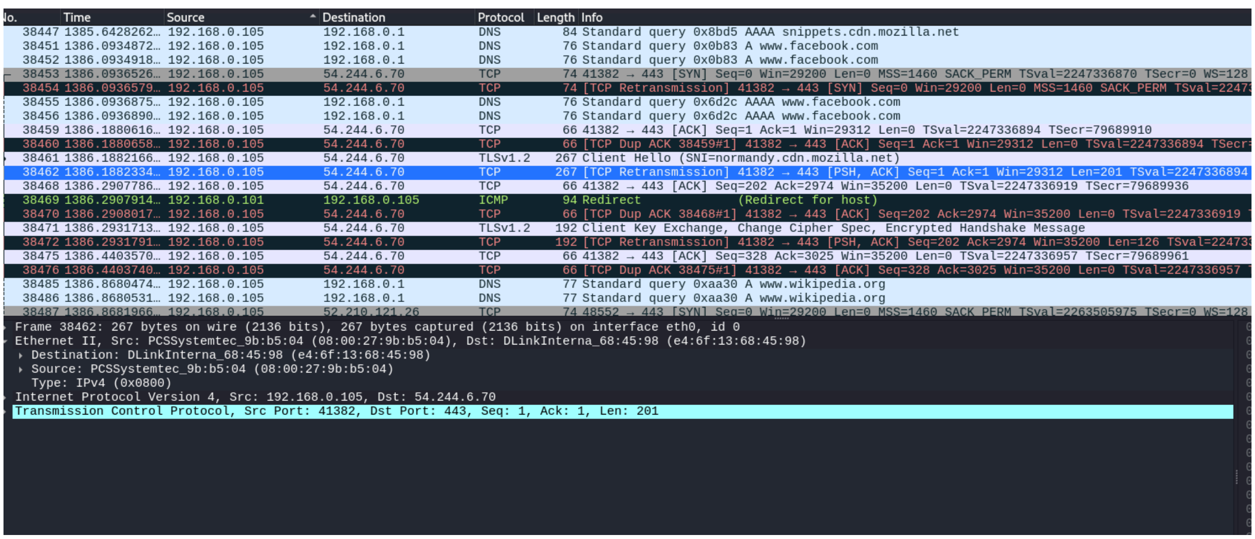


From the two screenshots we can see that 192.168.0.101 tries to make 192.168.0.105 map 192.168.0.1 to his mac address 08:00:27:9b:b5:04 instead of the right one e4:6f:13:68:45:98 by sending falsified arp packet

After that we can see from the coming two screenshots that 192.168.0.101 performed man in the middle attack and intercepted the packets between 129.168.0.1 and 192.168.0.105



In the highlighted packet the destination is 192.168.0.105 and we see that the src mac address is 08:00:27:9b:b5:04 which is the mac address of the attacker 192.168.0.101 although the src ip address is not 192.168.0.101 which means this packet sent to 192.168.0.101 first then it forwarded it to the right destination 192.168.0.105



In the highlighted packet the src is 192.168.0.105 and we see that the dst mac address is 08:00:27:9b:b5:04 which is the mac address of the attacker 192.168.0.101 although the dst ip address is not 192.168.0.101 which means this packet sent to 192.168.0.101 first then it forwarded it to the right destination 54.244.6.70

Vulnerability Exploited:

- Lack of Authentication in ARP: The Address Resolution Protocol (ARP) does not require authentication. Any device on the network can send an ARP reply or request, and other devices will generally accept these messages without any verification of their truthfulness. This allows an attacker to send spoofed ARP messages.

2. ARP flooding

No.	Time	Source	Destination	Protocol	Length	Info
62619	2238.0447328…	PCSSystemtec_9b:b5:…	PCSSystemtec_4f:41:…	ARP	42	Who has 192.168.0.105? Tell 192.168.0.101
62597	2227.8048339…	PCSSystemtec_9b:b5:…	DLinkInterna_68:45:…	ARP	42	Who has 192.168.0.1? Tell 192.168.0.101
62496	2186.8537228…	PCSSystemtec_9b:b5:…	PCSSystemtec_4f:41:…	ARP	42	Who has 192.168.0.105? Tell 192.168.0.101
62472	2176.6050393…	PCSSystemtec_9b:b5:…	DLinkInterna_68:45:…	ARP	42	Who has 192.168.0.1? Tell 192.168.0.101
62293	2135.3888305…	PCSSystemtec_9b:b5:…	PCSSystemtec_4f:41:…	ARP	42	Who has 192.168.0.105? Tell 192.168.0.101
61597	1998.4287861…	PCSSystemtec_9b:b5:…	PCSSystemtec_4f:41:…	ARP	42	Who has 192.168.0.105? Tell 192.168.0.101
61474	1967.7087468…	PCSSystemtec_9b:b5:…	DLinkInterna_68:45:…	ARP	42	Who has 192.168.0.1? Tell 192.168.0.101
61400	1947.2287322…	PCSSystemtec_9b:b5:…	PCSSystemtec_4f:41:…	ARP	42	Who has 192.168.0.105? Tell 192.168.0.101
61366	1934.6847670…	PCSSystemtec_9b:b5:…	DLinkInterna_68:45:…	ARP	42	Who has 192.168.0.1? Tell 192.168.0.101
61258	1911.1328049…	PCSSystemtec_9b:b5:…	PCSSystemtec_4f:41:…	ARP	42	Who has 192.168.0.105? Tell 192.168.0.101
61224	1900.8927603…	PCSSystemtec_9b:b5:…	DLinkInterna_68:45:…	ARP	42	Who has 192.168.0.1? Tell 192.168.0.101
60600	1879.1328355…	PCSSystemtec_9b:b5:…	PCSSystemtec_4f:41:…	ARP	42	Who has 192.168.0.105? Tell 192.168.0.101
60395	1858.6527544…	PCSSystemtec_9b:b5:…	DLinkInterna_68:45:…	ARP	42	Who has 192.168.0.1? Tell 192.168.0.101
60303	1847.3888906…	PCSSystemtec_9b:b5:…	PCSSystemtec_4f:41:…	ARP	42	Who has 192.168.0.105? Tell 192.168.0.101
60155	1827.9328825…	PCSSystemtec_9b:b5:…	DLinkInterna_68:45:…	ARP	42	Who has 192.168.0.1? Tell 192.168.0.101
60085	1815.3887465…	PCSSystemtec_9b:b5:…	PCSSystemtec_4f:41:…	ARP	42	Who has 192.168.0.105? Tell 192.168.0.101
59565	1796.1887495…	PCSSystemtec_9b:b5:…	DLinkInterna_68:45:…	ARP	42	Who has 192.168.0.1? Tell 192.168.0.101
59468	1783.3886976…	PCSSystemtec_9b:b5:…	PCSSystemtec_4f:41:…	ARP	42	Who has 192.168.0.105? Tell 192.168.0.101
58968	1762.9087389…	PCSSystemtec_9b:b5:…	DLinkInterna_68:45:…	ARP	42	Who has 192.168.0.1? Tell 192.168.0.101
58950	1751.1329420…	PCSSystemtec_9b:b5:…	PCSSystemtec_4f:41:…	ARP	42	Who has 192.168.0.105? Tell 192.168.0.101
58537	1716.5727508…	PCSSystemtec_9b:b5:…	PCSSystemtec_4f:41:…	ARP	42	Who has 192.168.0.105? Tell 192.168.0.101

From this screenshot we can see that 192.168.0.101 is sending many many arp request packets to all devices in the network which may be a sign for Dos attack (arp flooding attack)

Vulnerability Exploited:

- Limited Size of ARP Cache: Network devices store ARP entries in a limited-sized cache. Each entry maps IP addresses to MAC addresses.

3. TCP scanning

No.	Time	Source	Destination	Protocol	Length	Info
33947	581.106733791	192.168.0.101	192.168.0.105	TCP	58	54788 → 1066 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
33941	581.096498064	192.168.0.101	192.168.0.105	TCP	58	54788 → 1755 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
33935	581.086206510	192.168.0.101	192.168.0.105	TCP	58	54788 → 625 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
33929	581.076035020	192.168.0.101	192.168.0.105	TCP	58	54788 → 1060 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
33923	581.065720065	192.168.0.101	192.168.0.105	TCP	58	54790 → 524 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
33917	581.012672347	192.168.0.101	192.168.0.105	TCP	58	54789 → 3814 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
33911	581.002253124	192.168.0.101	192.168.0.105	TCP	58	54789 → 13456 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
33905	580.989700771	192.168.0.101	192.168.0.105	TCP	58	54789 → 5815 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
33899	580.979609336	192.168.0.101	192.168.0.105	TCP	58	54789 → 407 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
33891	580.965420321	192.168.0.101	192.168.0.105	TCP	58	54789 → 524 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
33885	580.909921954	192.168.0.101	192.168.0.105	TCP	58	54788 → 3814 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
33879	580.899665095	192.168.0.101	192.168.0.105	TCP	58	54788 → 13456 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
33873	580.889171914	192.168.0.101	192.168.0.105	TCP	58	54788 → 5815 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
33867	580.877180629	192.168.0.101	192.168.0.105	TCP	58	54788 → 407 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
33859	580.865241636	192.168.0.101	192.168.0.105	TCP	58	54788 → 524 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
33851	580.820425707	192.168.0.101	192.168.0.105	TCP	58	54789 → 2701 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
33846	580.810172812	192.168.0.101	192.168.0.105	TCP	58	54789 → 8254 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
33840	580.799806983	192.168.0.101	192.168.0.105	TCP	58	54789 → 1083 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
33834	580.780078677	192.168.0.101	192.168.0.105	TCP	58	54789 → 2393 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
33828	580.763992341	192.168.0.101	192.168.0.105	TCP	58	54789 → 20031 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
33819	580.719840627	192.168.0.101	192.168.0.105	TCP	58	54788 → 2701 [SYN] Seq=0 Win=1024 Len=0 MSS=1460

35428	586.742028227	192.168.0.105	192.168.0.101	TCP	60	5907 → 54789 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
35426	586.731779008	192.168.0.105	192.168.0.101	TCP	60	2135 → 54789 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
35424	586.721647928	192.168.0.105	192.168.0.101	TCP	60	34572 → 54789 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
35422	586.711222537	192.168.0.105	192.168.0.101	TCP	60	1031 → 54789 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
35420	586.701262595	192.168.0.105	192.168.0.101	TCP	60	1036 → 54789 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
35418	586.641179678	192.168.0.105	192.168.0.101	TCP	60	5907 → 54788 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
35416	586.630932266	192.168.0.105	192.168.0.101	TCP	60	2135 → 54788 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
35414	586.620529606	192.168.0.105	192.168.0.101	TCP	60	34572 → 54788 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
35412	586.610322795	192.168.0.105	192.168.0.101	TCP	60	1031 → 54788 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
35410	586.600098155	192.168.0.105	192.168.0.101	TCP	60	1036 → 54788 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
35408	586.588541581	192.168.0.105	192.168.0.101	TCP	60	1151 → 54789 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
35406	586.570437307	192.168.0.105	192.168.0.101	TCP	60	5000 → 54789 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
35404	586.551857692	192.168.0.105	192.168.0.101	TCP	60	2394 → 54789 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
35402	586.539361199	192.168.0.105	192.168.0.101	TCP	60	19315 → 54789 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
35400	586.529123638	192.168.0.105	192.168.0.101	TCP	60	32772 → 54789 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
35398	586.518880688	192.168.0.105	192.168.0.101	TCP	60	2047 → 54789 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
35396	586.508801100	192.168.0.105	192.168.0.101	TCP	60	49157 → 54789 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
35394	586.497824725	192.168.0.105	192.168.0.101	TCP	60	1164 → 54788 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
35392	586.487718215	192.168.0.105	192.168.0.101	TCP	60	1151 → 54788 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
35390	586.469853608	192.168.0.105	192.168.0.101	TCP	60	5000 → 54788 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
35388	586.449866706	192.168.0.105	192.168.0.101	TCP	60	2394 → 54788 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

From the last screenshots we can see that 192.168.0.101 performs TCP SYN scanning on 192.168.0.105 but 192.168.0.105 reply that nearly all ports are aborted

Vulnerability Exploited:

- Predictable Response to Connection Attempts: TCP (Transmission Control Protocol) hosts must respond in specific ways to incoming connection requests, depending on the state of the port (open, closed, or filtered).

5)Prevention and Mitigation Strategies
• Suggest methods and tools that could potentially prevent such attacks
• Discuss possible network configuration changes and monitoring strategies that could be implemented

ARP Spoofing
There are a great many free tools8 designed to detect this type of attack (see Arpwatch, Nast, Snort, Patriot NG, ArpON, etc) that generate alerts when an abnormal use of the ARP protocol is detected. Look at the output that Arpwatch generates when changes are detected in ARP/IP assignments. In the case of Snort, this has a prefix processor ARP designed to generate alerts in the case of an ARP Spoof Attack, To activate it, you must uncomment the following line in snort.conf: #preprocessor arpspoof then add the IP/MAC pairs to the machines that you want to monitor so that the prefix processor observes an ARP packet where the IP address of the sender coincides with one of the added entries and the MAC address of the sender does not coincide with that saved, Snort generates an alert. To add an entry to snort.conf write: preprocessor arpspoof_detect_host: 192.168.254.254 00:0e:0c:c6:c5:82.
Another focus of attention for administrators is the search for cards that are functioning in a disordered way, which is quite common in this type of scenario. Tools such as Neped, Sentinel, AntiSniff or SniffDet are quite useful as they detect cards in this state.

DOS Attacks

There are a great number of DDoS attacks, in addition to those previously mentioned: Direct Attacks, TTL expiry attack, IP unreachable attack, ICMP transit attacks, Reflection Attacks, etc. They are very difficult to contain, especially when it involves a high volume of traffic. 13 Owning devices that enable you to stop these attacks are expensive, making contacting the ISP the most appropriate action. However, when the DDoS attack is not that excessive, an appropriate configuration of the operating system and affected service could help to counteract the attack. Some of these parameters can be found in /etc/sysctl.conf:
• tcp_syncookies: protects you against Syn Flood attacks (like the one described above).
• ignore_broadcasts: One type of DDoS attack are the well known Smurf attacks in which ICMP (echo request) packets are sent to a broadcast address with a false IP source. This false IP is the target of the attack, as it receives multiple echo reply response packets as a result of the broadcast packet sent by the attacker. One way of deactivating the ICMP echo-broadcast requests is by activating the following option:
sysctl -w net.ipv4.icmp_echo_ignore_broadcasts=1
• rp_filter: Known also as source route verification, it has the same purpose as Unicast RPF (Reverse Path Forwarding) 14 and uses Cisco routers. It is used to check that the packets that enter via an interface are accessible based on the source address, making it possible to detect IP Spoofing: sysctl -w net.ipv4.conf.all.rp_filter=1
For attacks that are performed by programs like LOIC, it is also possible to implement measures using iptables and hashlimit modules to limit the number of packets that you want a particular service to accept.

TCP Scanning

1. Firewall Configuration: Configure firewalls to block or rate-limit incoming connection attempts, especially from external sources.
2. Intrusion Detection/Prevention Systems (IDS/IPS): Deploy IDS/IPS solutions to detect and block malicious network activities, including port scanning attempts.
3. Port Knocking: Implement port knocking techniques where legitimate users must send a sequence of connection attempts to specific ports before gaining access to services.
4. Rate Limiting:Implement rate limiting mechanisms to limit the number of connection attempts per source IP address or per port.

Network Configuration Changes:

1. Service Hardening**:
   - Disable unnecessary services and ports on servers and network devices to reduce the attack surface. Only enable services that are required for operation and regularly update and patch them to mitigate known vulnerabilities.

2. **Port Randomization**:

- Configure network services to listen on non-standard ports or implement port randomization techniques. This can make it harder for attackers to predict which ports are open and reduce the effectiveness of port scanning attacks.

Monitoring Strategies:

1. **Network Traffic Analysis**:
   - Monitor network traffic patterns and analyze logs for signs of port scanning activity. Look for unusual spikes in connection attempts or repetitive connection patterns from the same source IP address.

2. **Log Analysis**:
   - Monitor system logs for unusual connection attempts, failed login attempts, or other indicators of potential attack activity. Regularly review and analyze logs to identify security incidents.

3. **Anomaly Detection**:
   - Implement anomaly detection mechanisms to identify abnormal network behavior, such as sudden increases in SYN packets or unexpected port scanning activity. Anomaly detection tools can help identify and alert on suspicious behavior in real-time.

4. **Honeypots**:
   - Deploy honeypot systems within the network to attract and detect malicious activity, including port scanning attempts. Honeypots can provide valuable insights into attacker tactics and techniques without exposing production systems to risk.

By implementing a combination of these methods and tools, organizations can effectively mitigate TCP scanning attacks and reduce the risk of exploitation of vulnerabilities like predictable responses to connection attempts. Regular security assessments and updates to security measures are also essential to stay ahead of evolving threats.

6)Timeline and Scenario Illustration

• Draw a timeline of the attacks as you understand them

• Sketch the network diagram showing the attack paths and methods