Q1: Why are higher port numbers (> 1024) still of interest for, e.g., hackers or penetration testers?
Higher port numbers (>1024) are still of interest for hackers or penetration testers because they may be associated with less well-known or less secure services or applications that may have vulnerabilities that can be exploited.

Many common services, such as HTTP (port 80) and HTTPS (port 443), run on well-known ports that are often protected by firewalls and other security measures. However, other services and applications may use non-standard or less commonly used ports that are not as well-protected.

Hackers and penetration testers may scan for open ports on a target system to identify these less common services and applications and then attempt to exploit any vulnerabilities they find. Additionally, some attackers may use higher port numbers to evade detection by security measures that may be more focused on monitoring well-known ports.

Therefore, it is important for system administrators and security professionals to monitor all open ports on their systems, regardless of their number, to ensure that all services and applications are properly secured and protected from potential attacks.

Q3: Some scans take a long time to complete. Why?! Look at the responses from the system and try to explain! (The answer has been mentioned during a lecture.)
Some scans can take a long time to complete for a variety of reasons, including:

Network Latency: Scanning tools typically communicate with the target systems through the network, and network latency can affect how long it takes for a scan to complete. Slow network connections or congestion can result in delays in data transmission, causing the scan to take longer to complete.

Scan Depth: The depth of the scan can also impact how long it takes to complete. Deep scans that require more thorough analysis of each target system may take longer than simple or superficial scans.

Size of Network: The size of the network being scanned can also affect how long a scan takes. Larger networks with many systems will take longer to scan than smaller networks.

System Resources: The resources available on the scanning system and target systems can also impact the speed of the scan. If the scanning system or target systems are under high load, the scan may take longer to complete.

Type of Scan: Different types of scans may take longer to complete than others. For example, vulnerability scans that require more in-depth analysis may take longer to complete than port scans that simply check for open ports.

Target System Configuration: The configuration of the target system can also affect the time it takes for a scan to complete. Systems with robust security measures or intrusion detection systems may take longer to scan than systems with minimal security measures.

In general, the complexity of the scan and the size of the network being scanned are the primary factors that determine how long a scan will take to complete. It is important to consider these factors and allocate sufficient time and resources to complete the scan.

Q4: Nmap can report ports to be in different states. What do the states "open", "closed", "filtered" and "unfiltered" mean?

Nmap is a popular network exploration and security auditing tool that can be used to scan and map networks for open ports and services. Nmap can report ports to be in different states, including "open", "closed", "filtered" and "unfiltered". Here is what each of these states means:

Open: A port is considered "open" if an application or service is actively listening and responding to Nmap probes. This means that data can be sent and received through that port.

Closed: A port is considered "closed" if there is no application or service listening on that port. This means that data cannot be sent or received through that port.

Filtered: A port is considered "filtered" if Nmap is unable to determine whether the port is open or closed because the network traffic to that port is being blocked by a firewall, router, or other network device.

Unfiltered: A port is considered "unfiltered" if Nmap can determine that the port is not being filtered by any network device, but Nmap is still unable to determine whether the port is open or closed.

In summary, "open" means that there is an active service or application on that port, "closed" means that there is no active service or application on that port, "filtered" means that the port is being blocked by a network device, and "unfiltered" means that the port is not being blocked, but Nmap is still unable to determine the state of the port.

Which are the most/least interesting ones?

The most or least interesting port states depend on the context of the network being scanned and the objectives of the scan. Here are some examples:

In a penetration testing scenario, "open" ports are generally considered the most interesting, as they can be used to gain access to the target system. "Closed" ports may also be of interest, as they can provide information about the target system's security posture.

In a network monitoring or intrusion detection scenario, "filtered" ports may be the most interesting, as they can indicate that a firewall or other network device is blocking traffic to a specific port, which may be an attempt to hide the presence of a service or application.

In a vulnerability scanning scenario, "open" ports may be of interest if they are associated with vulnerable services or applications. "Filtered" and "unfiltered" ports may also be of interest, as they may indicate the presence of a firewall or other network device that could be exploited to gain access to the target system.

In general, "unfiltered" ports are considered the least interesting, as they do not provide much information about the target system's security posture.

It's important to note that the most or least interesting port states can vary depending on the specific objectives of the scan and the context of the network being scanned. Therefore, it's important to carefully consider the objectives of the scan and the potential risks and vulnerabilities of the target system when interpreting Nmap scan results.

Q5: Null and Xmas scans

Null and Xmas scans are types of port scans used by network administrators and security professionals to identify potential vulnerabilities in a target system's security. These scans are called "stealth scans" because they attempt to avoid detection by the target system's intrusion detection system (IDS) or firewall.

A Null scan is a type of scan where the scanning tool sends a TCP packet with no flags set (i.e., all flags are set to 0) to the target system's ports. If the target system's port is closed, it will respond with a TCP RST packet. However, if the port is open, it will not respond at all, indicating that the port is "stealthed". This can indicate that the target system has a firewall or IDS in place that is blocking the Null scan.

An Xmas scan is similar to a Null scan, but instead of sending a TCP packet with no flags set, it sends a packet with the FIN, URG, and PUSH flags set. If the target system's port is closed, it will respond with a TCP RST packet. However, if the port is open, it will not respond at all, indicating that the port is "stealthed". This can indicate that the target system has a firewall or IDS in place that is blocking the Xmas scan.

Both Null and Xmas scans are considered less reliable than other types of port scans, such as SYN scans or TCP connect scans, because they rely on the behavior of the target system's TCP/IP stack. Additionally, some modern IDS and firewall systems are able to detect and block Null and Xmas scans. Therefore, it is important to carefully consider the objectives of the scan and the potential risks and vulnerabilities of the target system before using these types of scans.

Null scans and Xmas scans send out specific types of TCP packets to target systems. Here's a breakdown of what they send out:

Null scan:

- A TCP packet is sent with all flags set to 0 (i.e., no flags set).
- This packet is sent to a target system's port to determine whether it is open, closed, or filtered.
- If the target system's port is closed, it will respond with a TCP RST packet.
- If the target system's port is open, it will not respond at all.

Xmas scan:

- A TCP packet is sent with the FIN, URG, and PUSH flags set.
- This packet is sent to a target system's port to determine whether it is open, closed, or filtered.
- If the target system's port is closed, it will respond with a TCP RST packet.
- If the target system's port is open, it will not respond at all.

What are the similarities and differences between the five scan types, i.e., syn, ack, fin, null, and Xmas? What are their use-cases, i.e., in which situation can each scan be useful?

The five main types of port scans used by network administrators and security professionals are SYN scan, ACK scan, FIN scan, Null scan, and Xmas scan. Each type of scan has its own characteristics and use cases. Here's a breakdown of their similarities and differences, as well as their use cases:

SYN scan:

Sends a SYN packet to the target system's port.

- If the port is open, the target system will respond with a SYN/ACK packet.
- If the port is closed, the target system will respond with a RST packet.
- If the port is filtered, the target system will not respond at all.
- SYN scan is the most commonly used and reliable port scan.
- It can be useful in identifying open ports and potential vulnerabilities.

ACK scan:

- Sends an ACK packet to the target system's port.
- If the port is filtered, the target system will respond with a RST packet.
- If the port is open or closed, the target system will not respond at all.
- ACK scan is used to determine whether a firewall is in place.
- It can be useful in identifying open ports and potential firewall rules.

FIN scan:

- Sends a FIN packet to the target system's port.
- If the port is open, the target system will not respond at all.
- If the port is closed, the target system will respond with a RST packet.
- If the port is filtered, the target system will not respond at all.
- FIN scan is used to determine whether a firewall is in place.
- It can be useful in identifying open ports and potential firewall rules.

Null scan:

- Sends a packet with all flags set to 0 to the target system's port.
- If the port is open, the target system will not respond at all.
- If the port is closed, the target system will respond with a RST packet.
- If the port is filtered, the target system will not respond at all.
- Null scan is used to determine whether a firewall is in place.
- It can be useful in identifying open ports and potential firewall rules.

Xmas scan:

- Sends a packet with the FIN, URG, and PUSH flags set to the target system's port.
- If the port is open, the target system will not respond at all.
- If the port is closed, the target system will respond with a RST packet.
- If the port is filtered, the target system will not respond at all.
- Xmas scan is used to determine whether a firewall is in place.
- It can be useful in identifying open ports and potential firewall rules.

In summary, SYN scan is the most commonly used and reliable port scan, while ACK, FIN, Null, and Xmas scans are used to determine whether a firewall is in place and can be useful in identifying open ports and potential firewall rules.

However, it is important to use these scans with caution and to carefully consider the objectives of the scan and the potential risks and vulnerabilities of the target system before using them.

Q6: How is UDP scanning done by Nmap? Why is this type of scan more problematic than TCP scans?

Don't underestimate UDP. It has a huge (and growing) presence on networks today. Critical services like DHCP, DNS, and VoIP have been running over UDP for decades. But more and more we are seeing emerging protocols such as QUIC and even RDP over UDP. Let's learn how to analyze these protocols.

UDP scanning is done by Nmap by sending UDP packets to the target system's ports and analyzing the responses. Here's how it works:

- Nmap sends a UDP packet to the target system's port.
- If the target system responds with a UDP packet, the port is considered open.
- If the target system responds with an ICMP packet (such as "Destination Unreachable") or does not respond at all, the port is considered closed or filtered.

UDP scanning is more problematic than TCP scanning for several reasons:

- UDP is a connectionless protocol, meaning that there is no handshake process between the scanning system and the target system. This makes it difficult to determine whether a port is open, closed, or filtered.
- Many network devices, such as firewalls and routers, are configured to drop UDP packets instead of responding with an ICMP packet. This can cause false positives or false negatives in the scan results.
- UDP scanning can also generate a large amount of traffic, which can trigger intrusion detection systems (IDS) and alert system administrators.

Due to these challenges, UDP scanning requires a careful approach and is often used as a complementary scan to TCP scanning. It is important to consider the potential risks and vulnerabilities of the target system before performing a UDP scan and to use caution to avoid triggering IDS or causing network disruptions.

Q7: How much time does it take to perform a UDP scan on the systems? Is there a difference between the systems in their responses? Explain!

The time it takes to perform a UDP scan on a system can vary depending on several factors, such as the number of ports being scanned, the network speed and congestion, and the responsiveness of the target system. UDP scanning can be slower than TCP scanning because UDP is a connectionless protocol and there is no handshake process to establish a connection.

In terms of differences in system responses, UDP scanning can be more challenging than TCP scanning because UDP packets are often dropped by network devices, such as firewalls and routers, instead of being responded to with an ICMP packet. This can result in false positives or false negatives in the scan results. Some systems may also have UDP ports that are not intended to be publicly accessible, which can cause concerns about security risks and vulnerabilities.

It is important to carefully consider the potential risks and challenges of UDP scanning before performing it on a system, and to use caution to avoid triggering intrusion detection systems (IDS) or causing network disruptions. Performing a scan on a limited number of ports at a time and using tools such as Nmap's "timing" options can also help to minimize the impact of the scan on the target system and network.

TCP fingerprinting

Use Nmap to figure out what kind of operating system the target machines run.

TCP fingerprinting is a technique used to identify the operating system and/or applications running on a remote target system by analyzing the unique characteristics of its TCP/IP protocol stack. This is achieved by sending a series of TCP packets with various flags set, and analyzing the responses received from the target system.

The TCP fingerprinting process typically involves sending a series of specially crafted packets to the target system, with various TCP flags set in different combinations. These packets are designed to trigger specific responses from the target system, which can be analyzed to identify the operating system or applications running on the system.

Nmap is a commonly used tool for TCP fingerprinting, using a technique known as OS detection. Nmap sends a series of TCP packets with various flag combinations and analyses the responses to determine the target system's operating system. Nmap also includes a database of fingerprints for common operating systems and applications, which can be used to identify the target system's software based on its response to the packets.

TCP fingerprinting can be useful for network administrators and security professionals to identify potential security risks and vulnerabilities in their networks. However, it can also be used by attackers to gather information about target systems and plan targeted attacks. As a result, it is important for system administrators to be aware of the risks and take appropriate security measures to protect their networks.

Q8: What packet(s) does Nmap send to figure this out?

In general, Nmap uses a combination of SYN, ACK, FIN, PSH, and URG flags in its packets to gather information about the target system's TCP/IP stack. For example, Nmap may send a SYN packet to determine if the system responds with a SYN-ACK packet, indicating that the system is likely running a Unix-based operating system. Nmap may also send packets with certain options and payload sizes to determine the presence of specific applications and protocols running on the target system.

The exact packets and flags used by Nmap can be customized using various command-line options and arguments, such as the -sT (TCP connect scan) and -sS (TCP SYN scan) options. The specific flags and packet configurations used by Nmap can also be customized using the tool's "fingerprint database" and "service and version detection" features.

Fragmentation

Q9: Why would you want to use fragmented packets for scanning?

Fragmentation is a technique used in network communication to break up large data packets into smaller fragments that can be transmitted more efficiently across the network. This is particularly useful in cases where the maximum transmission unit (MTU) of a network is smaller than the size of the data packet being transmitted.

In the context of network scanning and reconnaissance, fragmentation can be used to evade network intrusion detection systems (IDS) and firewall filters that may be configured to block or flag certain types of network traffic. By fragmenting packets, an attacker can split up the payload of the packets into smaller pieces, making it more difficult for IDS and firewall filters to detect and block the traffic.

Nmap, a popular network scanning and reconnaissance tool, includes the option to perform fragmented packet scanning using the -f (fragment packets) command-line option. This option instructs Nmap to send fragmented packets with the "don't fragment" (DF) bit set in the IP header, which can help to evade certain types of network filters and defences.

It is worth noting, however, that fragmentation can also introduce additional network latency and processing overhead and can potentially cause some network devices to drop or reject fragmented packets altogether. As a result, the use of fragmentation in network scanning and reconnaissance should be approached with caution and used only when necessary and appropriate for the specific circumstances.

How do you think fragmented packets should be handled in modern networks?

In modern networks, fragmented packets should be handled with care, and it is important to have appropriate network security policies and controls in place to detect and prevent any potential security threats associated with fragmented packets.

One approach to handling fragmented packets is to implement network-based security controls, such as intrusion detection and prevention systems (IDPS), that are specifically designed to detect and block fragmented packets that are associated with network attacks or reconnaissance activities. These security controls should be regularly updated and tuned to ensure that they are effective in detecting and blocking potential threats associated with fragmented packets.

Another approach is to apply best practices for network security, including implementing network segmentation, access control, and encryption protocols that help protect against network attacks and data breaches. Additionally, regular network vulnerability assessments and penetration testing can help to identify and address any potential security weaknesses or vulnerabilities associated with fragmented packets.

Overall, the handling of fragmented packets in modern networks requires a comprehensive and multi-layered approach to network security, including a combination of technical controls, security policies, and user awareness and training. By taking a proactive and comprehensive approach to network security, organizations can help protect against potential security threats associated with fragmented packets and other types of network attacks.

Q10: When checking the output with Wireshark, do you see fragmented packets as expected?

When checking the output of a network scan or reconnaissance tool with Wireshark, fragmented packets may be visible in the captured network traffic if the tool was configured to send fragmented packets. However, the visibility and interpretation of fragmented packets in Wireshark can depend on a number of factors, including the version of Wireshark being used, the specific packet capture settings, and the nature and characteristics of the fragmented packets themselves.

In general, fragmented packets may appear in Wireshark as a series of smaller packets with the "More Fragments" flag set in the IP header, indicating that there are additional fragments that are part of the original data packet. Wireshark may also display information about the original packet size, the number of fragments, and the offset of each fragment in relation to the original packet.

However, the visibility and interpretation of fragmented packets in Wireshark can be complicated by factors such as network congestion, packet loss, and retransmission of packets, which can affect the order and timing of packet fragments. Additionally, some network devices or security controls may be configured to block or discard fragmented packets altogether, which may affect their visibility in Wireshark.

Overall, while fragmented packets may be visible in Wireshark, their appearance and interpretation in the captured network traffic can be complex and dependent on a variety of factors, and may require additional analysis and interpretation to fully understand their significance in the context of a network scan or reconnaissance activity.

Q11: If you are going to remember only one thing from this lab, what should it be?

If you are going to remember only one thing about network scanning and reconnaissance, it should be the importance of using these techniques only for legitimate and authorized purposes, and with the appropriate level of permission and oversight. Network scanning and reconnaissance can be powerful tools for understanding and assessing network security, but they can also be used for malicious purposes such as network attacks and data breaches.