

# Week 1 and week 2

This comprehensive study and revision guide is designed to help you master the core concepts covered in your assignment and corresponding lecture material for the Ethical Hacking course. We will draw connections between cybersecurity fundamentals and essential networking principles.

---

## Comprehensive Study and Revision Guide

### Module 1: Foundations of Ethical Hacking

#### Topic 1.1: Ethical Hacking and Penetration Testing (Pen Test)

| Section                 | Content Explanation  |
|-------------------------|--|
| Overview                | Ethical hacking involves locating weaknesses and vulnerabilities in computer and information systems by replicating the intent and actions of malicious hackers, but doing so legally and with permission. <b>It is also known as penetration testing, intrusion testing, or red teaming.</b>  |
| Key Concepts            | <b>Objective:</b> The main objective is to <b>legally identify system vulnerabilities.</b> This process helps organizations strengthen their defenses. Ethical hackers are employed by companies to perform penetration tests.   |
| Related Terminology     | <b>Hacking</b> refers generally to showing computer expertise. <b>Cracking</b> involves breaching security on software or systems. <b>Security Test</b> is a broader term that includes analyzing a company's security policy and procedures, and <b>the tester offers solutions</b> to secure the network. In contrast, a pure <b>Penetration Test</b> is a legal attempt to break into the network to find weak links, but the tester only reports findings and <b>does not provide solutions.</b> |
| Cyber Attack Techniques | Techniques used by malicious actors include: <b>Spoofing</b> (faking the originating IP address), <b>Denial of Service (DoS)</b> (flooding a host with traffic so it cannot respond), <b>Port Scanning</b> (searching for vulnerabilities), and exploiting <b>Trojan Horses</b> (malware usually hidden inside downloaded software that may install backdoors).  |

**Relationships Between Topics (M1.1)** Ethical hacking is essentially professional, authorized penetration testing. Understanding the actions of a hacker (e.g., modifying logs, installing back

doors, performing DoS attacks) is crucial for the ethical hacker whose job is to identify and preempt these weaknesses.

### Revision Pointers (M1.1)

- Distinguish clearly between the objective of ethical hacking (legal identification of vulnerabilities) versus malicious hacking (stealing information, deleting files).
- Memorise the difference between a Penetration Test (report findings only) and a Security Test (report findings and offer solutions).
- Be familiar with basic hacker actions (e.g., DoS, Spoofing, Trojan Horses).

### Practice and Reflection Prompts (M1.1)

1. Why is obtaining explicit legal permission essential before beginning any penetration testing activity? (Reflection on legal responsibility).
2. If an ethical hacker discovers an unknown software vulnerability, should they immediately patch it, or just report it? Explain using the definitions of penetration testing vs. security testing.
3. Explain how **Time to Live (TTL)** (from Module 3) acts as a defense mechanism against a packet-based Denial of Service attack.

## Topic 1.2: Penetration Testing Methodologies

| Section                | Content Explanation   |
|------------------------|---|
| <b>Key Concept</b>     | Penetration testing methodologies define the level of information provided to the tester prior to the engagement.   |
| <b>White Box Model</b> | The tester has <b>complete information</b> about the network topology and technology. The tester may also be authorized to interview IT personnel and company employees, making the job easier. |
| <b>Black Box Model</b> | The tester <b>does not have any information</b> about the network details. The burden is entirely on the tester to discover the details.  |
| <b>Gray Box Model</b>  | This is a <b>hybrid</b> approach. The company provides the tester with <b>partial information</b> about the network.  |
| <b>Other Tools</b>     | <b>Tiger Box</b> refers to a collection of operating systems (OSs) and hacking tools, usually on a laptop, used by testers to conduct vulnerability assessments and attacks.                    |

**Relationships Between Topics (M1.2)** These methodologies determine the starting point of the ethical hacking exercise. Black box testing simulates an external attacker with no prior knowledge, while white box testing simulates an internal attacker or security audit with full access to internal systems.

## Revision Pointers (M1.2)

- Be able to define and differentiate the three primary methodologies (White, Black, Gray) based on the level of network information provided.
- Remember that "Red Box" is not a type of penetration testing methodology mentioned.

## Practice and Reflection Prompts (M1.2)

1. A company wants to test its network defenses against a determined external attacker with no insider knowledge. Which testing model should they choose, and why?.
  2. If a tester uses the Gray Box model, what type of partial information might they typically receive (e.g., network segment maps, specific application names)? (Reflection Prompt).
  3. List the pros and cons of using the White Box model compared to the Black Box model from the perspective of time and comprehensiveness.
- 

## Module 2: Network Fundamentals: Architecture and Switching

### Topic 2.1: Network Switching Techniques

| Section                     | Content Explanation   |
|-----------------------------|---|
| Circuit Switching           | A <b>dedicated communication path</b> (or fixed sequence of links) is established between two stations before data transfer begins. This path is reserved for the entire duration of the communication.   |
| Circuit Switching Concepts  | Requires three steps: Connection establishment (initial delay), data transfer (at maximum speed), and connection termination (to deallocate resources). It is <b>acceptable for voice communication</b> but <b>very inefficient for bursty traffic like data</b> .                      |
| Packet Switching            | The modern form of long-distance data communication. Network resources are <b>not dedicated</b> , meaning a link can be shared (dynamic bandwidth). Data is transmitted in short <b>packets</b> (chunks of a longer message, each containing a header for routing).                     |
| Packet Switching Concepts   | Based on the <b>store-and-forward concept</b> . Each intermediate node (router) receives the whole packet, decides the route (using a routing table), and forwards it. It is <b>more efficient for bursty data traffic</b> because it allows multiple communications over shared links. |
| Packet Switching Approaches | 1. <b>Virtual Circuits (VC):</b> Similar to circuit switching, a <b>route is established a priori</b> , and all packets follow the same path, but the links are <b>not dedicated</b> . 2. <b>Datagram:</b> No route is established beforehand;  |

each packet is transmitted as an independent entity, and nodes take routing decisions dynamically.

**Relationships Between Topics (M2.1)** Packet switching overcomes the inefficiency of circuit switching for data traffic by allowing links to be shared. The two packet switching approaches (Virtual Circuit and Datagram) trade off reliability/sequencing (VC) for flexibility/speed (Datagram).

### Topic 2.2: Datagram Packet Switching

| Section                    | Content Explanation   |
|----------------------------|---|
| <b>Key Characteristics</b> | Datagram packets are <b>independent</b> and use <b>dynamic routing</b> . They <b>do not require prior route establishment</b> . They may follow <b>different paths</b> between two hosts. |
| <b>Efficiency</b>          | Datagram switching is <b>faster for fewer packets</b> because it avoids the initial delay associated with route establishment and termination.  |
| <b>Drawbacks</b>           | Since paths are independent and dynamically routed, packets may be delivered <b>out of order</b> . If a node crashes, queued packets are lost, and duplicate packets may be generated.    |

### Revision Pointers (M2.1 & M2.2)

- Focus on the core differences: Dedicated vs. Shared links (Circuit vs. Packet).
- Contrast Virtual Circuit (fixed path, non-dedicated links) vs. Datagram (dynamic routing, independent packets).
- Understand why packet switching (specifically Datagram) is suitable for bursty data traffic and why it's faster for smaller transactions.

### Practice and Reflection Prompts (M2.1 & M2.2)

1. A large file is broken into 50 packets. If Datagram switching is used, what is the major risk the destination host must handle?.
2. Textually describe the concept of a virtual circuit as it relates to packet switching. How does it differ structurally from a circuit-switched network?.
3. Why does message switching incur higher delay compared to packet switching? (Hint: consider the store-and-forward approach and bandwidth utilization).

### Topic 2.3: Layered Network Architecture (OSI and TCP/IP)

| Section | Content Explanation |
|---------|---------------------|
|---------|---------------------|

|   |   |
|---|---|
| <b>OSI Model</b>                            | The Open Systems Interconnection (OSI) reference model is a conceptual <b>seven-layer model</b> . The purpose of layering is a systematic design approach where changes in one layer should not require changes in other layers.    |
| <b>TCP/IP Model</b>                         | The TCP/IP model does not strictly follow OSI; it uses a <b>simplified four-layer model</b> . This suite of protocols is fundamentally built on top of <b>connectionless technology (datagrams)</b> .                               |
| <b>Network Layer Functions (OSI/TCP-IP)</b> | <b>Network Layer (Layer 3):</b> Responsible for establishing, maintaining, and terminating connections. Crucially, it <b>routes packets</b> across point-to-point links. Devices working here are <b>Routers/Layer-3 Switches</b> . |
| <b>Data Link Layer Functions</b>            | <b>Data Link Layer (Layer 2):</b> Handles framing and error detection. Ensures reliable transfer of frames over a point-to-point link (flow control, error control). Devices working here are <b>Bridge/Layer-2 Switches</b> .      |
| <b>Transport Layer Functions</b>            | <b>Transport Layer (Layer 4):</b> Provides <b>end-to-end reliable data transfer</b> with error recovery and flow control. It is responsible for providing reliability in transmission.  |

**Relationships Between Topics (M2.3)** The IP layer (Network Layer, Layer 3) is responsible for routing. However, IP itself provides an unreliable, connectionless service. TCP (Transport Layer, Layer 4) builds on top of IP to add reliable, connection-oriented service, handling error control and reassembly.

### Revision Pointers (M2.3)

- Know the core responsibility of the IP layer: routing and addressing.
- Be able to map reliability/error control to the correct layer (Transport Layer/TCP).
- Understand that framing is handled by the data-link layer.

### Practice and Reflection Prompts (M2.3)

1. Explain the concept of data encapsulation in TCP/IP. What happens to the data chunk as it moves from the Application layer down to the Datalink layer?.
  2. If a packet arrives at a router (a Layer 3 device), what information in the packet header is most important for the router to perform its function?.
  3. If a non-standard protocol is used, which layers in the OSI model remain largely unaffected conceptually? (Reflection on Layer Independence).
-

## Module 3: Internet Protocol (IP) Layer Details

### Topic 3.1: IP Datagram Structure and Fields

| Section                         | Content Explanation  |
|---------------------------------|--|
| <b>IP Layer Role</b>            | The IP layer provides a connectionless, unreliable delivery system for packets (datagrams). Each IP packet must contain the source and destination addresses.  |
| <b>Header Size</b>              | The IP layer prepends a header of <b>minimum 20 bytes</b> to the data chunk received from the higher layer.  |
| <b>HLEN<br/>(Header Length)</b> | A <b>4-bit field</b> defining the header length, expressed as the number of 32-bit words. The minimum size is 5 (20 bytes) and the <b>maximum possible value is 15</b> (1111 in binary). If HLEN = 15, the header size is $15 \times 4 = 60$ bytes.                            |
| <b>TTL (Time to Live)</b>       | An <b>8-bit field</b> that prevents infinite looping of packets. The value is decremented at each router hop, and when it reaches 0, the packet is discarded.  |
| <b>Header Checksum</b>          | A <b>16-bit wide field</b> used for header integrity. It covers only the IP header. If a mismatch occurs, the datagram is discarded. The calculation uses ones complement arithmetic.  |
| <b>Other Fields</b>             | <b>Total Length</b> (16 bits) is the length of the datagram (header + data) in bytes. <b>Protocol</b> (8 bits) identifies the higher layer protocol being used (e.g., TCP or UDP). Fields like Identification, Flags, and Fragment Offset are used for handling fragmentation. |
| <b>Invalid IP Header Fields</b> | <b>Port Number</b> is a field of the transport layer, and <b>MAC address</b> is added in the data-link layer. These are not valid fields of the IP header.   |

**Relationships Between Topics (M3.1)** The IP header fields are critical for the Network Layer function: TTL manages looping/discardng packets, the IP addresses manage routing, and HLEN determines where the data payload begins. These fields ensure packets can traverse the connectionless, unreliable IP infrastructure effectively.

### Revision Pointers (M3.1)

- Be able to calculate the maximum HLEN value and corresponding header size.
- Understand the function of TTL (preventing loops) and Header Checksum (ensuring header integrity).
- Know which fields belong to IP (TTL, HLEN, Source/Dest IP) and which belong to other layers (Port Number, MAC Address).

### Practice and Reflection Prompts (M3.1)

1. If an IP packet has a total length of 4000 bytes and the HLEN value is 10, calculate the number of data bytes in the packet.
2. If a router detects that an IP header checksum is incorrect, what action does the router take, and why?
3. Why is the maximum possible datagram size limited to 65536 bytes? (Hint: refer to the size of the Total Length field).

### Topic 3.2: IP Fragmentation

| Section                              | Content Explanation  |
|--------------------------------------|--|
| <b>Why Fragmentation is Needed</b>   | Fragmentation is required because the <b>Maximum Transfer Unit (MTU) across networks varies</b> . If a packet is larger than the MTU of a network it needs to pass through, it must be divided into smaller fragments.   |
| <b>Fragmentation Process</b>         | Fragmentation is typically done by <b>intermediate routers</b> . Each fragment is transmitted as a separate IP packet. Reassembly of the packets is performed by the <b>final destination host</b> . IP uses <b>non-transparent fragmentation</b> .  |
| <b>Non-Transparent Fragmentation</b> | Fragmentation is <i>not</i> transparent to subsequent networks. Fragments are not reassembled at intermediate routers. Each fragment is treated as an independent packet, and the final destination host handles reassembly. This allows multiple exit routers and achieves higher throughput.   |
| <b>Reassembly Fields</b>             | For reassembling fragmented packets at the final destination, the following IP header fields are used:<br><b>Identification (ID)</b> : A datagram ID set by the source. All fragments of the original packet share the same ID.<br><b>Flags</b> : A 3-bit field. The <b>M bit (More Fragment)</b> specifies if this fragment is the last one ( $M=0$ indicates the last fragment, $M=1$ indicates more fragments follow). The <b>D bit (Don't Fragment)</b> prevents fragmentation if set to 1.<br><b>Fragment Offset</b> : A 13-bit field indicating where, in the original datagram, this fragment belongs. Specified in multiples of 8 bytes. |

**Relationships Between Topics (M3.2)** Fragmentation is enabled by specific fields within the IP header (M3.1), allowing the unreliable IP layer to handle transmission across links with different physical constraints (MTUs). The destination host needs to utilize the Identification, Flags, and Offset fields together to correctly reconstruct the original data sequence.

### Revision Pointers (M3.2)

- Know *who* fragments (intermediate routers) and *who* reassembles (final destination host).
- Master the function of the M flag (More Fragment) and Offset field in identifying the segment's position.
- Understand the context: fragmentation happens due to varying MTUs.

### Practice and Reflection Prompts (M3.2)

1. An IP packet arrives with the M flag set to 1 and the Offset set to 0. Describe the packet's status.
2. If the Fragment Offset is 100, what is the starting byte position of this fragment in the original data stream? (Hint: Offset is specified in multiples of 8 bytes).
3. Why is the D bit (Don't Fragment) important in network diagnostics and path discovery tools? (Reflection Prompt).

### Topic 3.3: IP Addressing and Subnetting

| Section                          | Content Explanation   |
|----------------------------------|---|
| <b>IP Address Basics</b>         | An IP address is a <b>32-bit quantity</b> expressed in dotted-decimal notation (W.X.Y.Z). It consists of two logical parts: a <b>network number</b> (assigned by a central authority) and a <b>host number</b> (assigned by a local administrator).   |
| <b>Classful Addressing</b>       | There are five defined IP address classes (A, B, C, D, E). They are identified by the first few bits of the address: <b>Class A</b> (Starts with 0, Range 0–127); <b>Class B</b> (Starts with 10, Range 128–191); <b>Class C</b> (Starts with 110, Range 192–223); <b>Class D</b> (Multicast, Starts with 1110); <b>Class E</b> (Reserved).                             |
| <b>Special/Private Addresses</b> | Addresses reserved for private use include <b>10.x.x.x</b> (Class A), <b>172.16.x.x – 172.31.x.x</b> (Class B), and <b>192.168.x.x</b> (Class C). Other special addresses include Loopback (127.x.x.x) and Limited Broadcast (255.255.255.255).   |
| <b>Subnetting</b>                | Subnetting introduces a third level of hierarchy (Network portion, Subnet portion, Host portion) to the IP address structure. It allows for more efficient utilization of addresses and the creation of smaller subnets from one network.   |
| <b>Subnet Mask</b>               | A subnet mask is used to split the IP address. The mask is a series of contiguous 1's followed by contiguous 0's. Natural masks are fixed based on class (Class A: 255.0.0.0; Class B: 255.255.0.0; Class C: 255.255.255.0). The <b>Subnet Address</b> is found by performing a bit-by-bit <b>AND</b> operation between the Destination IP address and the Subnet Mask. |

|   |   |
|---|---|
| <b>VLSM (Variable Length Subnet Masks)</b>      | Allows the same network to be configured with <b>different masks</b> , enabling subnets of different sizes. This maximizes address utilization.   |
| <b>CIDR (Classless Internet Domain Routing)</b> | A modern concept that eliminates the traditional class (A, B, C) structure. An IP address is represented by a prefix followed by a slash and a number (M) indicating the number of leftmost contiguous bits used for the network mask (e.g., 144.16.192.57 / 18). <b>The number of addresses in each CIDR block must be a power of 2.</b> |

**Relationships Between Topics (M3.3)** Classful addressing provided the initial structure for IP organization. Subnetting refined this by allowing networks to be subdivided using masks. VLSM and CIDR represent advanced methods developed to overcome the wastage and inflexibility inherent in the fixed, original classful model, improving address management and routing efficiency.

### Revision Pointers (M3.3)

- Be able to identify the class of an IP address based on its first octet.
- Memorise the reserved private IP address ranges.
- Practice calculating the subnet address using the bit-by-bit AND operation.
- Understand the fundamental shift CIDR created (moving away from A/B/C).

### Practice and Reflection Prompts (M3.3)

1. Determine the class of the IP address 192.0.0.1 and explain how the number of available hosts differs from an address starting 128.x.x.x.
  2. Calculate the subnet address for IP 172.31.10.5 using the mask 255.255.240.0 (Explain the steps using binary AND).
  3. Why did the Internet shift from the Classful model to using CIDR notation, even though existing Class A/B/C networks can still be represented in CIDR format?.
- 

## Module 4: Transport Layer (TCP and UDP)

### Topic 4.1: TCP and UDP Fundamentals

| Section                     | Content Explanation  |
|-----------------------------|--|
| <b>Transport Layer Role</b> | The Transport layer protocols (TCP and UDP) sit above IP. User processes interact with the TCP/IP suite by sending/receiving data via TCP or UDP, which then use the IP layer for packet delivery. |

|  |   |
|--|---|
| <b>TCP<br/>(Transmission Control Protocol)</b> | Provides a <b>connection-oriented, reliable, full-duplex, byte-stream service</b> . It handles end-to-end reliability using checksums, positive acknowledgements, and timeouts. TCP also manages the establishment and termination of connections and sequences data that may arrive out of order.            |
| <b>UDP (User Datagram Protocol)</b>            | Provides a <b>connectionless, unreliable service</b> for sending datagrams. It is simpler and <b>faster than TCP</b> . UDP does not split data into multiple packets and <b>does not care about error control</b> . It is often used for messages small enough to fit in a single packet (e.g., DNS queries). |
| <b>UDP Header Size</b>                         | The UDP header size is <b>8 bytes</b> (composed of 4 fields, each 16 bits).   |

**Relationships Between Topics (M4.1)** TCP and UDP both rely on the unreliable IP layer (Module 3) for basic delivery. TCP adds complex mechanisms (sequence numbers, ACKs) to achieve reliability, whereas UDP provides efficiency by skipping these reliability checks, making it ideal for low-latency applications.

#### Topic 4.2: Port Numbers and Connections

| Section   | Content Explanation  |
|---|--|
| <b>Port Numbers</b>                                   | Both TCP and UDP use <b>16-bit integer port numbers</b> . These numbers uniquely identify the specific <b>user processes (applications)</b> on a host that are sending or receiving data. Port numbers are stored in the TCP or UDP headers. |
| <b>Connection Identification</b>                      | A unique process-to-process connection is called an <b>association</b> , defined by a set of five values: the protocol, Local IP address, Local Port number, Remote IP address, and Remote Port number.                                      |
| <b>Well-Known Ports</b>                               | Predefined and publicly known ports (e.g., FTP uses port 21, SMTP uses port 25). These are generally reserved in the range 1 to 1023 (or extended to 4095).  |
| <b>Ephemeral Ports</b>                                | Temporary port numbers requested by a client process from the local TCP/UDP module. They are used for the client side of a connection and generally use numbers beyond the reserved range (up to 65535).                                     |
| <b>TCP Connection Establishment (3-way Handshake)</b> | A connection request is initiated when a segment is sent with <b>SYN=1 and ACK=0</b> . Connection confirmation (the second step) is represented by <b>SYN=1 and ACK=1</b> .  |

## Revision Pointers (M4.1 & M4.2)

- Be able to clearly contrast TCP and UDP regarding reliability, connection status, and speed.
- Understand the function of the 16-bit Port Number (identifying processes) and the difference between well-known and ephemeral ports.
- Know the flag values (SYN/ACK) used during the TCP 3-way handshake.

## Practice and Reflection Prompts (M4.1 & M4.2)

1. Explain why TELNET (a connection-oriented protocol) is unsuitable for real-time video streaming where slight packet loss is acceptable.
  2. A malicious actor floods a server with initial TCP connection requests (SYN=1, ACK=0) but ignores the server's SYN+ACK response. What type of attack is this, and why is it effective?.
  3. If a UDP packet arrives at a host, why does the host still need the port number even though UDP is connectionless?.
- 

## Module 5: Key Takeaways and Final Review

This module synthesizes critical conceptual knowledge across the topics studied.

| Concept Area                 | Key Takeaways & Core Relationships   |
|------------------------------|--|
| Hacking & Network Layer (L3) | Ethical Hacking seeks to exploit network weaknesses. Understanding Layer 3 (IP) is crucial, as IP is connectionless and unreliable. This lack of reliability and connection state is necessary for global routing but leaves open avenues for misuse (like spoofing IP addresses).   |
| Reliability Stack            | Reliability is built upward: The unreliable IP Layer handles routing. Reliable transport (TCP) is layered <i>on top</i> of IP, adding reliability through sequencing, acknowledgements, and error control. UDP bypasses this complexity for speed.                                   |
| Addressing System Evolution  | The rigid classful addressing system (A, B, C) led to resource inefficiency. This drove the creation of Subnetting (using masks to divide networks) and ultimately <b>CIDR</b> (classless routing) and <b>VLSM</b> (variable subnet sizes), improving address allocation efficiency. |
| Packet Control               | IP headers contain mechanisms for traffic control: <b>TTL</b> prevents routing loops, <b>Header Checksum</b> ensures the header wasn't corrupted in transit, and <b>Fragmentation fields</b> allow packets to adapt to varying MTUs across links.                                    |

## Final Revision Checklist

| Topic                      | Checkpoint  | Source Reference |
|----------------------------|---|------------------|
| <b>Ethical Hacking</b>     | Define objective; Differentiate Pen Test vs. Security Test.   |                  |
| <b>Pen Testing Methods</b> | Define White, Black, and Gray Box models.   |                  |
| <b>Switching</b>           | Contrast Circuit vs. Packet switching; know which is efficient for bursty data.                     |                  |
| <b>Datagrams</b>           | Know key characteristics: dynamic routing, faster for fewer packets, risk of out-of-order delivery. |                  |
| <b>IP Header Fields</b>    | Function and size of HLEN, TTL, and Header Checksum (16 bits).                                      |                  |
| <b>IP Addressing</b>       | Identify IP address classes (A, B, C); list private address ranges.                                 |                  |
| <b>Subnetting</b>          | Calculate subnet address using IP and Subnet Mask (binary AND operation).                           |                  |
| <b>Fragmentation</b>       | Know why it is needed (MTU); identify fields for reassembly (ID, Offset, Flags).                    |                  |
| <b>TCP vs. UDP</b>         | Contrast connection status, reliability, speed, and typical use cases.                              |                  |
| <b>TCP Handshake</b>       | Identify flag states for connection request (SYN=1, ACK=0) and confirmation.                        |                  |
| <b>Port Numbers</b>        | Know their size (16 bits) and function (process identification).                                    |                  |

## Week 3 and week 4

This comprehensive study and revision guide is designed to help you prepare effectively for your exams by synthesizing key concepts from the assignment solutions and lecture slides concerning Routing, IPv6, and Information Gathering in Ethical Hacking.

---

# Comprehensive Study and Revision Guide: Networking, IPv6, and Ethical Hacking Foundations

## Section 1: Networking and Routing Fundamentals

This section covers the basic methods used for packet movement across networks, including connection types and delivery mechanisms.

### 1.1 Connection Options: Connection-Oriented vs. Connection-Less

| Concept                    | Explanation & Key Concepts (R-1, R-6)  |
|----------------------------|--|
| <b>Connection-Oriented</b> | The network layer first establishes a connection. All subsequent packets are delivered according to this pre-established connection.   |
| <b>Connection-Less</b>     | The network layer treats each packet independently. <b>There is no relationship between packets.</b> The IP protocol uses a <b>connection-less approach</b> for packet delivery. |

**Relationships and Clarification (R-2, R-6):** The IP protocol, which forms the foundation of the Internet, relies on connection-less routing. This means that statements claiming IP uses connection-oriented routing are false.

#### Revision Pointers (R-3):

- Differentiate clearly between the handling of packets in each approach.
- Remember that IP (both IPv4 and IPv6) is fundamentally connection-less.

#### Practice and Reflection Prompts (R-4):

1. Explain the primary benefit of IP using connection-less routing rather than connection-oriented routing.
2. If a sequence of five packets is sent using a connection-less protocol, and one packet is lost, how are the remaining four packets affected?

### 1.2 Packet Delivery Options: Direct vs. Indirect

| Concept | Explanation & Key Concepts (R-1) |
|---------|----------------------------------|
|---------|----------------------------------|

|                          |  |
|--------------------------|--|
| <b>Direct Delivery</b>   | Occurs when the source and destination of the packet are <b>located on the same physical network</b> . This includes host-to-host delivery on the same network or delivery between the last router and the destination host. |
| <b>Indirect Delivery</b> | Used when the destination host is <b>not on the same network</b> as the deliverer. Involves the packet travelling through <b>multiple routers</b> until it reaches the final router connected to the destination network.    |

**Relationships (R-2):** Indirect delivery relies on routing methods (see 1.3) to guide the packet across various networks and routers.

#### **Revision Pointers (R-3):**

- Know the physical network constraints for Direct Delivery.
- Recognise that "router to router" movement is the defining feature of Indirect Delivery.

#### **Practice and Reflection Prompts (R-4):**

1. A router needs to send a packet to a host connected directly to one of its interfaces. Is this direct or indirect delivery? Why?
2. If a packet travels from H1 → R1 → R2 → H2 (where R2 connects to H2's network), identify where the indirect delivery ends and the direct delivery begins (summarizing the network topology diagrammatically).

## **1.3 Routing Methods and Tables**

#### **Routing Methods (R-1, R-6):**

1. **Next-hop routing:** Routing tables are based on identifying the *next hop* towards the destination.
2. **Network-specific routing:** Routing tables use the destination *network address*.
3. **Host-specific routing:** Allows specification of a route to a single host address.
4. **Default routing:** Used as a **gateway of last resort** for forwarding packets whose destination address does not match any existing entry in the routing table. The **default route in IPv4 CIDR notation is 0.0.0.0/0**.

#### **Routing Table Types: Static vs. Dynamic (R-1, R-6):**

| Concept              | <b>Static Routing</b>  | <b>Dynamic Routing</b>  |
|----------------------|--|---|
| <b>Definition</b>    | Entries are <b>manually configured</b> by the network administrator. | Routes are <b>updated automatically</b> based on network condition. |
| <b>Changeability</b> | <b>Does not change with time</b> unless manually modified.           | Routing table <b>updates periodically</b> .                         |

|                       |   |   |
|-----------------------|---|---|
| <b>Resource Use</b>   | Lower resource consumption.                       | Routers <b>exchange control information, which consumes network resources.</b>      |
| <b>Link Failure</b>   | Administrator needs to manually update the table. | Tables can be updated automatically; can <b>automatically find alternate paths.</b> |
| <b>Protocols Used</b> | None (manual entry).                              | Uses protocols like RIP, OSPF, BGP, etc..   |

**Key Takeaways (R-5): Routing Table Fields and Flags** A routing table contains fields such as Subnet mask, Destination IP address, Next hop address, and Flags. The **G flag stands for Gateway**, indicating that the route uses an intermediate gateway (router) instead of being directly connected. Other flags include:

- **U:** Route is up and running.
- **H:** Specifies a host-specific route.
- **D:** Added by redirection (dynamically created).
- **M:** Modified by redirection.

**Relationships (R-2):** The need for automatic updates and path recovery during link failures (Dynamic Routing) is what necessitated the development of sophisticated **Routing Protocols** (Section 2).

#### **Revision Pointers (R-3):**

- Know the specific meaning of the **Static** routing table entry type.
- Identify the four correct characteristics of **Dynamic** routing.
- Memorise the function and name associated with the **G flag**.
- Know that the default route is specified by **0.0.0.0**.

#### **Practice and Reflection Prompts (R-4):**

1. Describe a scenario where Network-specific routing would be more efficient than Host-specific routing.
2. Why is dynamic routing preferred in large, complex networks despite its consumption of network resources?
3. If a routing table entry shows the flag sequence UG, what does this tell you about the status and configuration of that route?

## **Section 2: Internet Routing Protocols**

This section delves into how routing information is exchanged using specific protocols, classified by their operational boundary (Autonomous Systems).

## 2.1 Autonomous Systems (AS) and Protocol Classification

**AS Definition (R-1):** An Autonomous System (AS) is a set of routers and networks managed by a single organisation. Each AS is assigned a unique AS number.

**Protocol Classification (R-1, R-6):**

1. **Interior Routing Protocols (Intra-AS):** Used to exchange information between routers *within* the same AS. Examples include **RIP** and **OSPF**.
2. **Exterior Routing Protocols (Inter-AS):** Used to exchange routing information *between* routers in different AS's. The main example is **BGP**.

## 2.2 Routing Information Protocol (RIP)

**Key Concepts (R-1, R-6):**

- RIP is an **interior routing protocol**.
- It uses **distance vector routing** based on **hop count**.
- It updates table entries using values received from neighbours.
- **Problem:** RIP suffers from the **count-to-infinity problem**. This is a drawback of distance vector routing, where slow convergence for larger networks means it takes a long time for all routing tables to detect when a network becomes inaccessible. Routing loops may also take a long time to be detected.

## 2.3 Open Shortest Path First (OSPF)

**Key Concepts (R-1, R-6):**

- OSPF is a widely used **interior routing protocol** in TCP/IP networks.
- It updates routing tables based on **link state advertisements (LSA)**.
- It computes the least-cost route (based on configurable costs like delay, data rate) using **Dijkstra's algorithm**.
- **Hello packets** are sent every 10 seconds (configurable) to check if a neighbour router is up (active).
- The absence of a "Hello" packet for 40 seconds indicates a failure of the neighbour.
- **OSPF Packet Types** include Hello, Database Description, Link State Request, Link State Update (LSAs flooded), and Link State Acknowledgement (for reliable flooding).

## 2.4 Border Gateway Protocol (BGP)

**Key Concepts (R-1, R-6):**

- BGP is the **most widely used exterior router protocol** for the Internet.
- It allows routers belonging to **different autonomous systems** to exchange routing information.

- It uses **TCP on port number 179** to send routing messages.
- It is a distance vector protocol, but unlike RIP, BGP contains **complete routes**.
- Routers exchanging information are called **peers**.
- Initially, peers exchange the **entire routing table**, and subsequently, only incremental updates are sent.
- **BGP Message Types** include: **Open** (to open a connection), **Update** (to transmit route information/advertise new routes), **Keepalive** (to periodically confirm the connection is alive), and **Notification** (to report errors/close connection).
- BGP routers run by ISPs route packets to the outside world, often making the ISP router the default route for organizational networks.

### **Key Takeaways (R-5): Protocol Summary**

- **BGP** is the protocol used for routing **between different autonomous systems**.
- **RIP** uses distance vector routing and suffers from the **count-to-infinity** problem.
- **OSPF** uses **Hello Packets** to verify neighbour status.

**Relationships (R-2):** RIP and OSPF function internally within an AS, providing the routes needed for local traffic, while BGP handles the high-level routing *between* these independent AS structures. OSPF is generally considered an improvement over distance vector protocols like RIP due to its use of link state and Dijkstra's algorithm for faster convergence.

### **Revision Pointers (R-3):**

- Know the AS structure and the classification of protocols (Interior: RIP, OSPF; Exterior: BGP).
- Focus on the key drawbacks of RIP (count-to-infinity).
- Understand the fundamental mechanism of OSPF (Link State, Dijkstra, Hello packets).
- Identify BGP as the primary exterior protocol and remember its use of TCP/port 179.

### **Practice and Reflection Prompts (R-4):**

1. If a router is using RIP and one link fails, describe why it takes RIP longer than OSPF to fully update its routing tables across the network.
2. In a complex network environment, explain why an organization might run OSPF internally but rely on BGP only at the border routers connected to their ISP.
3. What is the purpose of a BGP Update message, and how does it differ from a Keepalive message?

## **Section 3: IP Version 6 (IPv6)**

This section details the next-generation IP protocol, its features, and how it addresses the limitations of IPv4.

### **3.1 Features and Design (R-1, R-6)**

**Problems with IPv4:** IPv4 suffers from a **limited address space** (32-bit address is inadequate) and lacks sufficient capabilities for real-time applications and complex addressing.

### Key Features of IPv6:

- **Address Size:** Uses **128-bit IP addresses**, providing a massive address space ( $2^{128}$  total addresses).
- **Connectionless:** Like IPv4, IPv6 is **connectionless**; each datagram is routed independently.
- **Header Format:** Uses a series of fixed-length headers. A datagram consists of a **base header followed by zero or more extension headers**.
- **Base Header Size:** The **base header size is 40 bytes**.
- **Addressing Classes:** Unlike IPv4, **IPv6 does not support address classes** like A, B, or C.
- **Hop Limit:** The header contains a Hop Limit, which is decremented by 1 at each hop; discarded when it reaches 0.
- **Notation:** Addresses are represented using **colon-hexadecimal notation** (groups of 16 bits written in hex, separated by colons).

### IPv6 Addressing Types:

- **Unicast:** Corresponds to a single computer.
- **Multicast:** Refers to a set of computers; packet is delivered to **every member of the set**.
- **Anycast:** Refers to a set of computers sharing the same address prefix; packet is delivered to **exactly one** of the computers in the set.

## 3.2 Transition Strategies (R-1)

To transition networks from IPv4 to IPv6, three main strategies are employed:

1. **Dual stack:** The gateway supports both IPv4 and IPv6 protocol stacks simultaneously.
2. **Tunneling:** An IPv6 datagram flows through an intermediate IPv4 network by encapsulating the whole IPv6 packet inside an IPv4 packet. This can be done automatically using IPv4-compatible IPv6 addresses (80 0's, 16 0's, followed by 32-bit IPv4 address).
3. **Header translation:** An IPv4 address is translated into an IPv6 address, and vice versa.

**Key Takeaways (R-5):** The false statement regarding IPv6 is that it supports address classes. Key specifications are the 128-bit address length and the 40-byte base header.

**Relationships (R-2):** IPv6 solves the crucial scalability issue (address exhaustion) faced by IPv4. The core connection method (connection-less routing) is retained from IPv4, emphasizing the foundational design principle of the Internet Protocol.

### Revision Pointers (R-3):

- Focus on the address size (128 bits) and base header size (40 bytes).
- Confirm that IPv6 *does not* use address classes.
- Distinguish between Unicast, Multicast, and Anycast delivery types.
- Understand the concept of Tunneling as encapsulation.

#### **Practice and Reflection Prompts (R-4):**

1. How does the introduction of Anycast addresses enhance service availability compared to traditional Unicast addressing?
2. If a network supports both IPv4 and IPv6, what is the simplest transition strategy they can employ, and how does it ensure communication across both protocols?
3. If an IPv6 packet needs to travel across a segment of the network that only uses IPv4 routers, which transition mechanism must be used?

## **Section 4: Ethical Hacking Environment Setup**

Before performing hacking activities, a controlled and isolated laboratory environment is essential.

### **4.1 Laboratory Setup Components (R-1, R-6)**

**Disclaimer:** It is **illegal to perform any kind of hacking activity** on vulnerable machines on the Internet/Intranet that do not belong to you. All demonstrations should be performed on victim systems installed in a virtual machine.

| <b>Component</b>                                | <b>Explanation and Role</b>   |
|---|---|
| <b>Hypervisor<br/>(Virtual Machine Monitor)</b> | Software that creates and runs <b>Virtual Machines (VMs)</b> . It allows <b>one host computer to support multiple guest VMs</b> by virtually sharing resources like memory, network interface, and storage. Examples: VMware, VirtualBox. |
| <b>Attacker System</b>                          | An OS containing penetration testing tools. <b>Kali Linux</b> (a Debian-based distribution) is highly recommended for beginners, containing thousands of tools for security research and penetration testing.                             |
| <b>Victim Machines</b>                          | Intentionally vulnerable (insecure and hackable) virtual machines designed for training and exploit testing. Examples: Metasploitable 2 (vulnerable Linux) and Metasploitable 3 (vulnerable Windows).                                     |

### **4.2 Network Setup in Virtual Box (R-1)**

- **Default Mode (NAT):** The Virtual OS is separated from the host system. The virtual box allocates virtual IPs and allows connection to the Internet.
- **Bridge Adapter Mode:** The best option for establishing a connection between the host system and other VMs installed inside the virtual box.

**Key Takeaways (R-5):** A hypervisor allows one host to run multiple virtual machines by sharing resources, making it essential for security practice. Kali Linux is a specific Linux distribution for penetration testing, not a "hack proof hypervisor".

**Relationships (R-2):** The Hypervisor provides the virtual sandbox necessary to isolate the attacker system (Kali Linux) from the real network while allowing safe interaction with the victim machine (Metasploitable).

#### **Revision Pointers (R-3):**

- Know the definition of a hypervisor and its function in resource sharing.
- Identify Kali Linux as the preferred attacker system.
- Recognize Metasploitable machines as intentionally vulnerable victims.

#### **Practice and Reflection Prompts (R-4):**

1. Why is it dangerous to skip the virtual machine setup and perform ethical hacking practice directly on production networks?
2. If you need your Kali Linux VM to communicate directly with your physical home router, which VirtualBox network mode should you use?
3. If a statement claims that a hypervisor requires one physical machine per virtual machine, why is this false?

## **Section 5: Information Gathering (Reconnaissance)**

Reconnaissance is the crucial first step in any attack, focusing on collecting information about a target network or system.

### **5.1 Passive vs. Active Reconnaissance (R-1, R-6)**

| Concept               | Passive Reconnaissance   | Active Reconnaissance   |
|-----------------------|--|---|
| <b>Communication</b>  | Information is collected <b>indirectly</b> .<br>The attacker and victim <b>do not communicate directly</b> . | Information is collected <b>directly</b> by communicating with the victim system. |
| <b>Detection Risk</b> | <b>No chance of detection.</b>   | <b>Chance of detection</b> due to direct communication (e.g., NMAP scan).         |

|                      |  |   |
|----------------------|--|---|
| <b>Detail Level</b>  | Provides publicly available data (e.g., web browsing). | Can provide more detailed information about the target machine. |
| <b>Tools/Methods</b> | Archive.org, Whois, Netcraft, Search Engines.          | NMAP, Nessus, DNS enumeration, Mail tracking.                   |

## 5.2 Passive Reconnaissance Tools (R-1, R-6)

Passive techniques involve gathering information without directly probing the target system.

- **Archive.org's Wayback Machine:** Used to view historical versions of a website. Useful for observing past webpage changes, mirroring the site, and observing directories.
- **WHOIS Lookup:** Provides ownership and registration details of a domain. Details include domain owner name, registrar, registration date, and expiration date.
- **Netcraft:** An internet service company used to find the list of subdomains and the operating system of the corresponding server.
- **Search Engine Operators (Google Dorking):** Used to restrict and refine searches for sensitive information:
  - " " (Double Quotes): Used to search for an exact phrase (that exact sequence of words).
  - site::: Limits the search results to only a specific website or domain.
  - cache::: Finds the most recent cache of a specified webpage.
  - filetype::: Filters results by a specific file type.

## 5.3 Active Reconnaissance: DNS and Mail Server Enumeration (R-1)

Enumeration involves creating active connections and performing directed queries to gain more information.

- **DNS/Mail Server Enumeration:** Locating DNS servers and their records.
- **Yields:** Usernames, computer names, IP addresses, and reveals the size of the organization.
- **Tools:** nslookup, host, dig.

**Key Takeaways (R-5):** Reconnaissance aims to collect network info (IPs, services), system info (usernames, routing tables), and organization info (employee details, security policies). Passive methods are undetectable, while active methods risk detection.

### Revision Pointers (R-3):

- Crucially distinguish between the risk/communication style of Passive vs. Active reconnaissance.

- Know the purpose of WHOIS and Archive.org.
- Memorise the functions of the `site:` and double quote (" ") operators.

#### **Practice and Reflection Prompts (R-4):**

1. If an attacker performs an NMAP scan, which type of reconnaissance is being carried out, and what is the associated risk?
2. How would you use Google Search Operators to find all PDF files published only on the `swayam.gov.in` domain?
3. Why might an attacker use the Wayback Machine instead of simply browsing the target's current website?

## **Section 6: Active Scanning (Network Mapper - NMAP)**

NMAP is a primary tool for active reconnaissance, used for host discovery and port scanning.

### **6.1 NMAP Fundamentals (R-1, R-6)**

NMAP (Network Mapper) is a free, open-source tool for **vulnerability scanning and network discovery**.

#### **Main Features:**

1. **Host Discovery:** Determining which hosts are alive.
2. **Port Scanning:** Identifying available services/open ports.
3. **Service and Version Detection:** Determining the specific version running.
4. **OS Detection:** Identifying the operating system.

**Default Behavior:** When no specific port option is provided, NMAP scans the **1,000 most commonly used ports** for TCP and UDP.

**Host Discovery Option:** The option `-sn` tells NMAP to **skip port scanning and perform only host discovery** (also known as ping scan).

### **6.2 Host Discovery Techniques (Ping Sweep) (R-1)**

Ping sweep is the operation of querying multiple hosts to detect which ones are live.

- **ICMP Sweep (-PE):** Sends an **ICMP ECHO request (Type 8)**. If an ICMP ECHO reply (Type 0) is received, the target is alive. It is easy to implement but easy to block.
- **TCP Sweep (-PS for SYN, -PA for ACK):** Sends TCP SYN or ACK packets to common ports (e.g., 21, 80). If a response is received, the target is alive.

- **UDP Sweep (-PU):** Sends a UDP datagram. If no ICMP PORT UNREACHABLE message is received, the target is considered alive. This method is less reliable as routers and firewalls often drop UDP packets.

### 6.3 Port Scanning Techniques (R-1)

Port scanning determines which services are running or LISTENING.

| Technique          | Mechanism   | Detection Risk   |
|--------------------|---|--|
| TCP Connect Scan   | Uses the <b>basic, complete TCP 3-way handshake</b> (SYN, SYN/ACK, ACK).  | Easy to detect by inspecting system logs.  |
| TCP SYN Scan (-sS) | Also known as <b>half-open scanning</b> . Sends SYN, receives SYN/ACK, but immediately terminates with <b>RST</b> instead of completing the handshake.  | Less detectable than Connect Scan, as the connection is never fully established. |
| TCP Stealth Scan   | Aims to avoid detection and logging by using "wrong packets" based on RFC793. Closed ports respond with RST; <b>open ports ignore the packet</b> (no response). Methods include FIN probe, Xmas probe, or RST scan. | Difficult to detect, requires long history logs.                                 |
| FTP Bounce Scan    | Connects to an FTP server and asks the server to initiate an active data transfer process to the target port.   | Quite slow.  |

### 6.4 NMAP Advanced Features and Options (R-1, R-6)

- **OS Detection (-O):** Enabled OS detection, relying on specific responses from the TCP/IP stack implementation (e.g., ACK values, TOS fields).
- **Service/Version Detection (-sV):** Probes open ports to determine service/version information.

### 6.5 Reconnaissance Countermeasures (R-1)

Organizations can prevent or mitigate reconnaissance efforts:

- Do not release critical information publicly.
- Use split DNS and restrict zone transfer.
- Disable directory listing.
- Examine logs for suspicious packets (e.g., connections not properly terminated).
- Encrypt passwords and sensitive information.

- Keep systems updated and use server masks.

**Key Takeaways (R-5):** NMAP is versatile, covering host discovery, port scanning, OS detection, and version detection. TCP SYN scan (-sS) is the most common stealth technique because it avoids full connection establishment.

**Relationships (R-2):** Active reconnaissance often relies entirely on tools like NMAP to perform scanning. The goal of port scanning (Section 6.3) is directly linked to the objective of reconnaissance: identifying running services and vulnerabilities.

#### **Revision Pointers (R-3):**

- Know the four core capabilities of NMAP.
- Remember the default number of ports NMAP scans (1000).
- Identify the difference between TCP Connect Scan (full handshake) and TCP SYN Scan (half-open, RST termination).
- Know the purpose of the -sn, -O, and -sV options.

#### **Practice and Reflection Prompts (R-4):**

1. An attacker wants to scan a target without leaving a full handshake record in the system logs. Which NMAP scan type should they use, and why is it called "half-open"?
2. If an NMAP output shows that port 80 is open, what additional NMAP flag would you use to confirm whether Apache or Nginx is running, and what its specific version is?
3. Why is an ICMP sweep often blocked by firewalls, and what alternative sweep method can an attacker use instead?

## **Week 5 and 6**

---

This comprehensive study and revision guide is designed to help you prepare effectively for exams based on the topics covered in your assignments and lecture slides concerning Ethical Hacking and Cryptography.

## **Comprehensive Study & Revision Guide: Ethical Hacking and Cryptography**

### **Module 1: Scanning, Reconnaissance, and Vulnerability Assessment**

This module focuses on the tools and techniques used early in the hacking lifecycle to discover systems and identify weaknesses.

## Topic 1.1: NMAP Scripting Engine (NSE) and Vulnerability Scanning

### 1. Explanation of Topic

- **Definition:** NMAP (Network Mapper) is a powerful tool used for network discovery and security auditing. The NMAP Scripting Engine (NSE) allows users to employ thousands of available scripts to perform various automated operations, including vulnerability detection, service discovery, and backdoor detection.
- **Key Concepts:**
  - **Default Script Location (Linux):** The default NSE script directory is `/usr/share/nmap/scripts`.
  - **Running Default Scripts:** The option `--script=default` runs the standard set of NSE scripts, which are designed for service discovery, version detection, and checking basic vulnerabilities.
  - **Vulnerability Testing Example:** The script `http-slowloris-check` is used to determine if a web server is vulnerable to a Slowloris DoS attack *without actually launching the DoS attack*. (Note: `http-Slowloris` is the script used to launch the actual attack).
- **Formulas/Diagrams (Textual Summary):**
  - To find scripts related to a specific keyword: `nmap --script "keyword-*"`.
  - To run a specific script: `nmap --script <script name> <port # if required> <target>`.
  - To list all scripts in Kali Linux: `ls -al /usr/share/nmap/scripts`.

### 2. Relationships Between Topics

- NMAP scripting is a subset of **Vulnerability Scanning**. Other dedicated tools like Nessus and OpenVAS are also used for vulnerability scanning.
- NMAP is critical for **Reconnaissance** (gathering information), which is a prerequisite for **System Hacking** (like password cracking).

### 3. Revision Pointers

- Know the exact path for default NMAP scripts on Linux.
- Distinguish between running the default script set (`--script=default`) and running all scripts (`--script=all`).
- Understand the difference between testing for a vulnerability (`http-slowloris-check`) and exploiting it.

### 4. Practice and Reflection Prompts

1. Why is running `--script=all` generally not recommended during a typical penetration test?

2. If you wanted to check for SMB-related operating system details, what is the correct NMAP script name, and how does this relate to the incorrect options listed in the assignment?
3. Explain the primary purpose of vulnerability scanning in ethical hacking.

## Topic 1.2: Dedicated Vulnerability Scanning Tools

### 1. Explanation of Topic

- **Definition:** Vulnerability scanning is the process of identifying known vulnerabilities and weaknesses in a system or network to determine potential exploitation vectors.
- **Key Concepts:**
  - **Nessus:** A popular remote security scanning tool. It scans a system/network and raises alerts if weaknesses are discovered. It is often preferred over NMAP by organizations for formal vulnerability assessment.
  - **Other Tools:** Besides NMAP and Nessus, tools like Nmapse, MBSA, and OpenVAS are also used for vulnerability scanning.
- **Key Takeaways (Nessus):** Nessus supports a wide variety of scanning options, offers an easy user interface, and generates detailed analysis reports. A key constraint for Nessus is that a free version is typically unavailable, requiring purchase, though a 7-day free trial might be offered.

### 2. Relationships Between Topics

- Tools like Nessus provide a comprehensive, organized approach to vulnerability scanning, whereas NMAP is often used for highly customized or targeted scans (using NSE scripts).
- Identifying vulnerabilities using Nessus or NMAP often precedes **Privilege Escalation** attempts, as design flaws or bugs found during scanning can be leveraged.

### 3. Revision Pointers

- Be able to name several dedicated vulnerability scanning tools (Nessus, Nmapse, OpenVAS).
- Know why Nessus is often favored by organizations compared to NMAP (more popular, detailed reports).

### 4. Practice and Reflection Prompts

1. Compare and contrast Nessus and NMAP concerning their popularity and general use case in large organizations.
2. What specific information might a vulnerability scan reveal that an initial NMAP port scan might miss?
3. If you were conducting a compliance audit, which tool might be more appropriate: Nessus or using only NMAP scripts, and why?

---

## Module 2: System Hacking and Password Attacks

This module covers gaining initial access to a system, typically by cracking login credentials, and the tools used in this process.

### Topic 2.1: Password Cracking Techniques and Tools

#### 1. Explanation of Topic

- **Definition (System Hacking):** The compromise of computer systems and software to gain unauthorized access to a target computer, often to steal or misuse stored information.
- **Definition (Password Cracking):** A set of techniques used to recover passwords from computer systems, often succeeding due to weak or easily guessable passwords.
- **Key Techniques (System Hacking):**
  - **Shoulder Surfing:** Observing the user entering their login details.
  - **Social Engineering:** Manipulating people into revealing secrets.
  - **Dictionary Attack:** Using a dictionary file (wordlist) against user accounts.
  - **Brute-Force Attack:** Trying every possible combination of characters until the password is broken.
  - **Rule-based Attack:** Used when some information about the password structure is known.
  - **Password Guessing:** Manually trying passwords compiled from collected information (e.g., social engineering).
  - **Wire Sniffing:** Capturing raw network traffic (using packet sniffer tools) which might contain clear-text usernames and passwords.
  - **Rainbow Table:** A precomputed table containing word lists and their corresponding hash values, used to bypass hashing protection.
  - **Keylogger/Trojan/Spyware:** Malicious software that runs in the background to send information back to the attacker.
- **Key Tools:**
  - **Crunch:** Used to **generate custom wordlists/password dictionaries** based on rules (like length, character sets, prefixes/suffixes).
  - **Hydra:** A fast and flexible login cracker used to perform **brute-force login attempts** on numerous network services and protocols (e.g., SSH, FTP, HTTP, SMB, Telnet).
  - **Other tools:** `john the ripper` and `hashcat`.

#### 2. Relationships Between Topics

- **Crunch** is preparatory; it creates the input file (wordlist) used by brute-forcing tools like **Hydra**.

- Password cracking is a crucial step in **System Hacking**. Success often leads to **Privilege Escalation** attempts, as initial access is often gained via a non-admin account.
- Using **Keyloggers** is a specific type of malware attack, which is part of the larger **Malware** category.

### 3. Revision Pointers

- Clearly differentiate between the function of Crunch (wordlist generator) and Hydra (brute-force executor).
- Understand the fundamental difference between dictionary and brute-force attacks.
- Recognize that tools like Wireshark (packet sniffer) and DNSEnum (user enumeration) are *not* brute-force tools like Hydra.

### 4. Practice and Reflection Prompts

1. Describe a scenario where a rule-based attack would be more efficient than a purely brute-force attack.
2. If an attacker successfully captures encrypted network traffic, which password cracking technique (from the list above) becomes more relevant, and why?
3. How can the principle of **User Enumeration** (identifying valid usernames on a target system/service) aid in setting up a targeted brute-force attack using Hydra?

## Topic 2.2: Privilege Escalation

### 1. Explanation of Topic

- **Definition:** Privilege escalation occurs after an attacker gains initial network access using a non-admin account. The goal is to gain higher, often administrative, privileges. This is achieved by exploiting design flaws, programming errors, bugs, or configuration oversights.
- **Consequences:** Elevated privileges allow the attacker to view sensitive information, delete files, or install malicious programs (viruses, Trojans, worms).
- **Types:**
  - **Vertical Privilege Escalation:** Gaining **higher** privileges than the existing account (e.g., standard user to administrator).
  - **Horizontal Privilege Escalation:** Acquiring the **same level** of privileges as already granted, but assuming the identity of another user with similar privileges.
- **Defences (Countermeasures):**
  - Restrict interactive logon privileges.
  - Use encryption techniques to protect sensitive data.
  - Run user-level applications with the least possible privileges.
  - Reduce the amount of code running with particular privileges.

### 2. Relationships Between Topics

- Successful password cracking often makes **Privilege Escalation** the attacker's next immediate step.
- Once privileges are escalated, an attacker can proceed with **Application Execution and File Hiding**, such as installing backdoors or keyloggers.

### 3. Revision Pointers

- Focus on defining and distinguishing between Vertical and Horizontal escalation.
- Remember the defense principle: *Principle of Least Privilege* (running applications on the least privileges).

### 4. Practice and Reflection Prompts

1. Give an example of a situation where horizontal privilege escalation might be more useful than vertical escalation for a social engineering attack.
  2. An attacker gains access to a machine and uses an overlooked system bug to gain root access. What type of privilege escalation is this, and why is defense using encryption alone insufficient to prevent this specific attack?
- 

## Module 3: Malware and System Compromise

This module details various malicious software used for system compromise and data theft.

### Topic 3.1: Malware Types (Keylogger, Trojan, Virus, Worm, Ransomware)

#### 1. Explanation of Topic

- **Definition (Malware):** Malicious software designed to damage or disable computer systems and grant limited or full control to the creator for theft or fraud.
- **Examples of Malware:** Trojan Horse, Rootkit, Ransomware, Adware, Virus, Worms, Spyware, Botnet.
- **Key Concepts & Types:**
  - **Keylogger:** Malware that monitors and **records every keystroke** typed on the keyboard, often used to steal sensitive information like passwords.
  - **Trojan Horse:** Malicious code contained inside seemingly harmless code or data; it gets activated by predefined user actions. Trojans can create covert communication channels for data transfer, generate fake traffic for DoS attacks, record screenshots, or create backdoors.
  - **Virus:** A self-replicating program that attaches itself to another program, boot sector, or document to produce a copy. They are characterized by infection stages and may encrypt or transform themselves.
  - **Worm:** Self-replicating programs, similar to viruses, often transmitted via file downloads or email attachments.

- **Ransomware:** Restricts access to a computer system's files and demands an online ransom payment for removal of restrictions. It is currently quite common.
- **Malware Entry Points:** Instant Messenger, IRC, Removable devices, Attachments, Browser/email software bugs, NetBIOS (File Sharing), Fake programs, Untrusted sites/freeware.

## 2. Relationships Between Topics

- A **Keylogger** is a specific type of malware that an attacker might execute remotely after achieving **Privilege Escalation**.
- Trojans can be used to generate fake traffic, which contributes to **Denial of Service (DoS) attacks**.

## 3. Revision Pointers

- Know the specific function of a Keylogger (keystroke recording).
- Understand how a Trojan differs from a Virus (Trojan disguises itself within harmless code; Virus self-replicates by attaching).
- Recognize that **Ransomware** is characterized by file restriction and ransom demand.

## 4. Practice and Reflection Prompts

1. If a user notices their system disabling their antivirus software and redirecting them to unknown pages, which type of malware might be indicated, and why?
  2. List three key countermeasures that can be taken against the spread of malware and Trojans.
  3. Why is updating software regularly considered an important defense against malware infection?
- 

## Module 4: Network Attacks (Sniffing, Spoofing, and DoS)

This module focuses on layer 2/3 network attacks that intercept or disrupt traffic flow.

### Topic 4.1: Packet Sniffing and ARP/MAC Attacks

#### 1. Explanation of Topic

- **Definition (Packet Sniffing):** The process of monitoring and capturing all data packets passing through a given network, acting as a form of wiretap. The Network Interface Card (NIC) is turned to **Promiscuous Mode** so it listens to all data transmitted on its segment.
- **Types of Sniffing:**

- **Passive Sniffing:** Occurs when monitoring traffic on a network connected via a **hub** (where traffic is sent to all ports). This is easy to execute but outdated as most networks use switches.
  - **Active Sniffing:** Used to sniff a **switch-based network**. It involves injecting ARP packets to flood the switch's Content Addressable Memory (CAM) table.
- **Vulnerable Protocols (Data sent in clear text):** HTTP, Telnet, Rlogin, POP, IMAP, SMTP, NNTP, FTP.
- **Key Protocol Attacks:**
  - **ARP (Address Resolution Protocol):** Used to map IP addresses to MAC addresses in a local network.
  - **MAC Attack / MAC Flooding:** Flooding the switch's fixed-size dynamic CAM table with fake MAC address/IP pairs until it is full. When full, the switch behaves like a hub, forwarding traffic to every port, enabling active sniffing.
  - **ARP Spoofing:** Forging ARP packets to send data to the attacker's machine, often to overload a switch or target computer's ARP cache.
  - **ARP Poisoning:** Flooding a target computer's ARP cache with forged entries using fake ARP messages. This diverts communication between two machines via the attacker's PC, facilitating threats like Packet Sniffing, Man-in-the-Middle Attacks, and Session Hijacking.
  - **DHCP Starvation Attack (DoS):** The attacker broadcasts forged DHCP requests to lease all available DHCP addresses in the scope, preventing legitimate users from obtaining or renewing an IP address.

## 2. Relationships Between Topics

- ARP Spoofing and MAC Flooding are methods used to enable **Active Sniffing** on switched networks.
- DHCP Starvation is a specific type of **Denial of Service (DoS)** attack that targets availability of IP resources.
- **Wire Sniffing** is explicitly mentioned as a method for password cracking.

## 3. Revision Pointers

- Understand the fundamental function of ARP (IP to MAC mapping).
- Know the purpose of promiscuous mode in packet sniffing.
- Distinguish between Passive (hub-based) and Active (switch-based, requiring MAC/ARP attacks) sniffing.
- Recognize that protocols sending data in clear text (e.g., HTTP, FTP, Telnet) are vulnerable to sniffing.

## 4. Practice and Reflection Prompts

1. Explain why a modern corporate network using switches requires an attacker to perform an Active Sniffing technique rather than relying on Passive Sniffing.

2. If you successfully perform ARP Poisoning, what major type of network attack does this enable you to execute against two communicating machines?
  3. How does a DHCP Starvation attack specifically compromise the security service of **Availability**?
- 

## Module 5: Cryptography Fundamentals and Security Services

This module introduces the basic concepts of security, attacks, services, and cryptographic primitives.

### Topic 5.1: Security Attacks and Services

#### 1. Explanation of Topic

- **Definition (Security Attacks):** Any action that compromises the security of information.
- **Four Primary Attack Types (Based on CIA Triad impact):**
  - **Interruption (Availability Attack):** An asset/service becomes inaccessible or unusable (e.g., a DoS attack).
  - **Interception (Confidentiality Attack):** An unauthorized party gains access to information (e.g., eavesdropping, wiretapping).
  - **Modification (Integrity Attack):** An unauthorized party tampers with data or a message.
  - **Fabrication (Authenticity Attack):** An unauthorized party inserts false data or messages. (Example: Masquerade attack).
- **Passive vs. Active Attacks:**
  - **Passive:** Obtain information being transmitted (eavesdropping). They are difficult to detect. Types include release of message contents and traffic analysis.
  - **Active:** Involve modification of the data stream or creation of a false stream. Categories include Masquerade (pretending to be someone else), Replay (capturing and replaying a transaction), Modification, and Denial of Service.
- **Key Security Services:**
  - **Confidentiality (Privacy):** Ensuring data is private.
  - **Authentication:** Verifying who created or sent the data.
  - **Integrity:** Ensuring data has not been altered.
  - **Non-repudiation:** Ensuring parties cannot later deny having participated in a transaction.
  - **Access Control:** Preventing misuse of resources.
  - **Availability:** Ensuring permanence and non-erasure.

#### 2. Relationships Between Topics

- DoS attacks (like DHCP Starvation or Slowloris) are categorized as **Interruption attacks**.

- Encryption is the **most important concept** for network security, and is the primary tool used to ensure **Confidentiality**.

### 3. Revision Pointers

- Link each of the four attack types (Interruption, Interception, Modification, Fabrication) to the security property they compromise (Availability, Confidentiality, Integrity, Authenticity).
- Understand the key difference: Passive attacks monitor; Active attacks modify or create data.

### 4. Practice and Reflection Prompts

1. If a Trojan generates fake traffic to overwhelm a server, which security attack category does this fall under, and what security service is being compromised?
  2. Explain why passive attacks are generally harder to detect than active attacks.
  3. What cryptographic primitive is often combined with encryption to achieve **Authentication**?
- 

## Module 6: Symmetric (Private) Key Cryptography

This module focuses on encryption where the same key is used for both encryption and decryption.

### Topic 6.1: Classical Ciphers and Key Management

#### 1. Explanation of Topic

- **Definition (Symmetric Key Cryptography):** Also known as Private Key Cryptography. A **single key (K)** is shared and used by both the sender and receiver for encryption and decryption.
- **Core Process:** The sender encrypts plaintext (P) using K to generate ciphertext (C):  $C = E(P, K)$ . The receiver decrypts C using K to get P:  $P = D(C, K)$ .
- **Security Principle:** The security depends solely on the **secrecy of the key**, not the algorithm, which is typically known to the public.
- **Classical Techniques:**
  - **Substitution Ciphers:** Each letter or group of letters is replaced by another letter or group.
    - **Caesar Cipher:** Replaces each letter with the  $k^{th}$  following letter (where  $k$  is the secret key). This is weak due to a small number of possibilities (25).
    - **Mono-alphabetic Cipher:** Allows arbitrary substitution. Though there are many possible keys (26!), it is easy to break by observing letter frequency.

- **Transposition Ciphers:** Letters are permuted (rearranged) in some form. The order of the column permutation becomes the key. The drawback is that the ciphertext retains the same letter frequency as the plaintext.
- **Key Distribution Problem:** For  $N$  parties communicating securely, a large number of keys are needed:  $N(N-1)/2$ . For 20 parties, 190 distinct key values are required.

## 2. Relationships Between Topics

- Classical ciphers demonstrate the core concept of **Symmetric Key Encryption** but are replaced by modern, complex ciphers (like DES and AES) because of their computational weaknesses.
- The large number of keys required for secure communication in symmetric systems highlights the practical advantage of **Asymmetric Key Cryptography** for key exchange.

## 3. Revision Pointers

- Know the formula for calculating the required number of keys for  $N$  parties in symmetric encryption ( $N C_2$ ).
- Be able to perform a simple substitution calculation (e.g., Caesar cipher shift  $k$ ).
- Understand the conceptual difference between substitution (replacement) and transposition (reordering).

## 4. Practice and Reflection Prompts

1. How many distinct key values would be required if 10 parties wanted to exchange messages securely using symmetric key cryptography?
2. If a mono-alphabetic cipher is used, what cryptographic attack technique is employed to break it, given its high number of possible keys?
3. Why is the security of symmetric encryption said to depend only on the key's secrecy, and not the algorithm?

## Topic 6.2: Modern Symmetric Algorithms (DES and AES)

### 1. Explanation of Topic

- **Practical Ciphers:** These are complex, require computers to execute, and can encrypt any kind of data (not just text). They are either **Stream Ciphers** (encrypt bit-by-bit) or **Block Ciphers** (encrypt  $n$ -bit blocks at a time).
- **Data Encryption Standard (DES):**
  - A widely used block cipher.
  - **Block size:** 64 bits.
  - **Key size:** 56 bits.
  - It uses a Feistel structure across 16 rounds.
  - Concern: The 56-bit key length is considered too short for critical applications.

- **Triple DES (3DES):** Uses three executions of the DES algorithm (Encrypt-Decrypt-Encrypt) with three keys (\$K\_1, K\_2, K\_3\$) to achieve an effective key length of 168 bits.
- **Advanced Encryption Standard (AES):**
  - Selected as the replacement for DES in 2000. Also known as Rijndael cryptosystem.
  - **Block size: 128 bits** (fixed by the AES standard).
  - **Key size: 128, 192, or 256 bits.**
  - It is resistant against known attacks and easy to implement.
  - AES typically uses 10, 12, or 14 rounds, involving steps like SubBytes, ShiftRows, MixColumn, and AddRoundKey.

## 2. Relationships Between Topics

- Both DES and AES are **symmetric key algorithms** that use the same key for encryption and decryption.
- The shift from DES to AES was motivated by the theoretical attacks on DES and its small 56-bit key size.
- Symmetric algorithms like AES are often used in conjunction with **Public Key algorithms (like RSA)**, where the public key system is used only to securely exchange the secret AES session key.
- Symmetric encryption (e.g., DES) is **much faster** (megabits/second) than asymmetric encryption (e.g., RSA) (kilobits/second)—about 100 times faster.

## 3. Revision Pointers

- Know the key lengths for DES (56 bits), Triple-DES (168 effective bits), and AES (128, 192, 256 bits).
- Be able to identify DES and AES as symmetric algorithms.
- Understand the fundamental trade-off: Symmetric keys are fast but suffer from the key distribution problem; Asymmetric keys are slow but solve key distribution.

## 4. Practice and Reflection Prompts

1. If a system uses AES, what is the mandatory block length, and what are the possible key lengths it might employ?
  2. Why was 3DES developed, and what specific problem with DES did it attempt to address?
  3. Explain the difference between a stream cipher and a block cipher, using AES as an example of one type.
- 

## Module 7: Asymmetric (Public) Key Cryptography

This module examines encryption methods using key pairs for different functions.

## Topic 7.1: Public Key Principles and RSA Algorithm

### 1. Explanation of Topic

- **Definition (Asymmetric Key Cryptography):** Also known as Public Key Cryptography. Uses **two keys** for every communication link: a Public Key ( $K_{UB}$ ) and a Private Key ( $K_{RB}$ ).
- **Key Concepts:**
  - **Key Generation:** Computationally easy for party B to generate a unique public/private key pair ( $K_{UB}$ ,  $K_{RB}$ ).
  - **Confidentiality (Encryption):** Sender A encrypts the message (P) using the receiver B's **Public Key** ( $K_{UB}$ ). Receiver B decrypts C using their own **Private Key** ( $K_{RB}$ ).
  - **Authentication (Digital Signature):** Sender A signs (encrypts) the message/hash using their own **Private Key** ( $K_{RA}$ ). Receiver B verifies the signature using Sender A's **Public Key** ( $K_{UA}$ ).
  - **Key Count:** For  $N$  parties, the number of distinct key values required is  $2N$  (one public and one private key per party). For 100 parties, 200 keys are required.
- **RSA Algorithm:** The most widely implemented block cipher, developed by Rivest, Shamir, and Adleman (1977).
  - **Security:** RSA's security relies on the computational difficulty of the **Prime Factorization Problem**—factoring the product ( $n$ ) of two very large prime numbers ( $p, q$ ). If an intruder knows  $p$  and  $q$ , they can find  $\Phi(n)$ , which helps them determine the private key ( $d$ ).
  - **Key Generation Steps (Textual Summary):** 1. Select two large primes,  $p$  and  $q$ . 2. Calculate  $n = p \times q$ . 3. Calculate Euler totient  $\Phi(n) = (p-1)(q-1)$ . 4. Select public exponent  $e$  (relatively prime to  $\Phi(n)$ ). 5. Calculate private exponent  $d = e^{-1} \pmod{\Phi(n)}$ . Public Key  $K_U = \{e, n\}$ , Private Key  $K_R = \{d, n\}$ .
  - **Encryption Formula:**  $C = M^e \pmod{n}$ .
  - **Decryption Formula:**  $M = C^d \pmod{n}$ .
  - **Key Size:** Keys are typically large, 1024 to 2048 bits, to ensure security.

### 2. Relationships Between Topics

- Asymmetric cryptography solves the **Key Distribution Problem** inherent in symmetric cryptography, as only the public key needs to be openly shared.
- Public key systems like RSA and Diffie-Hellman depend on complex **mathematical problems** (factorization and discrete logarithm, respectively) for their security.
- Decryption of a message in public-key cryptography requires the receiver's **Private Key**.

### 3. Revision Pointers

- Understand the key count formula for asymmetric systems ( $2N$ ).
- Know which key is used for encryption (Receiver's Public Key) and which is used for decryption (Receiver's Private Key).
- Identify the underlying difficult mathematical problem for RSA (Prime Factorization).
- Recognize that either key can be used for encryption, with the other used for decryption, enabling both confidentiality and authentication.

#### 4. Practice and Reflection Prompts

1. A receiver A wants to decrypt a message from B. Which key must A use for decryption, and why?
2. If quantum computers become widely available, which characteristic of the RSA algorithm is threatened, and what is the resulting area of research?
3. If 50 parties want to communicate securely using public-key cryptography, how many keys are needed in total?

### Topic 7.2: Diffie-Hellman Key Exchange and Message Authentication

#### 1. Explanation of Topic

- **Diffie-Hellman Key Exchange:** Proposed in 1976.
  - **Function:** Allows a group of users to agree on a **secret key over an insecure channel**.
  - **Limitation:** Cannot be used to directly encrypt and decrypt messages.
  - **Security:** Depends on the difficulty of computing **discrete logarithms**.
  - **Key Concept:** Requires no prior communication between parties A and B, who independently compute a shared secret key ( $k$ ).
  - **Vulnerability:** Susceptible to the **Man-in-the-Middle (MITM) Attack** by an active intruder.
- **Message Authentication:** Used to verify the integrity and origin of a message.
  - **Message Authentication Code (MAC):** A commonly used scheme where a MAC tag is generated as a function of the message (M) and a shared key (K):  $MAC = F(K, M)$ . The tag is appended to the message for transmission.
  - **Common Hashing Algorithms (for Authentication):** MD family (MD2, MD4, MD5 - 128 bit hash), SHA family (SHA-1, SHA-256, SHA-384, SHA-512), and RIPEMD family.

#### 2. Relationships Between Topics

- Diffie-Hellman addresses the problem of securely **exchanging a shared secret** (often a session key for a symmetric algorithm like AES) without requiring prior trust.
- Message authentication is used to guarantee **Integrity** and **Authenticity**. It can be done with or without conventional message encryption.

#### 3. Revision Pointers

- Know that Diffie-Hellman is for **key exchange**, not message encryption.
- Identify the weakness of the Diffie-Hellman algorithm (Man-in-the-Middle attack).
- Understand that MAC provides authentication by verifying that the message and key generate the same tag upon receipt.

#### 4. Practice and Reflection Prompts

1. What is the core mathematical problem that protects the security of the Diffie-Hellman Key Exchange?
2. If an attacker performs a Man-in-the-Middle attack against two parties using Diffie-Hellman, what must the attacker do to maintain secrecy while relaying messages?
3. Explain, using the concept of MAC, how a receiver can ensure the integrity of a message received over an insecure channel.

## Week 7 and 8

What a fantastic approach to revision! Preparing a comprehensive guide that links theory with practical examples is key to success. I'm delighted to put together this detailed study and revision guide for you, combining the precise details from the assignment solutions with the broader context provided in the lecture slides.

---

# Comprehensive Study and Revision Guide: Ethical Hacking Concepts

This guide is organized to provide a clear, logical progression through the topics, moving from foundational cryptographic tools to network protocols and common attacks.

## Module 1: Cryptographic Foundations and Authentication

### Topic 1.1: Cryptographic Hash Functions (MD5 and MAC)

| Section  | Explanation   |
|--|---|
| <b>Overview:</b><br><b>Definition &amp; Concepts</b> | Cryptographic hash functions are <b>computational functions</b> that take a message (\$M\$) of <b>arbitrary length</b> and generate a <b>fixed-length hash digest</b> (\$H\$). They are referred to as <b>one-way functions</b> , meaning it is not possible to uniquely retrieve \$M\$ from \$H\$. |

|  |   |
|--|---|
| <b>Theory: Key Properties</b>                  | Hash functions implement a <b>many-to-one mapping</b> . This means a very large input set maps to a finite output set. They must be <b>deterministic</b> (same input always gives the same output). Key desirable properties of cryptographic hash functions include: <b>Fixed-length output</b> , <b>One-way/Non-reversible</b> , and <b>Collision resistance</b> (it should be computationally infeasible to find two messages $x$ and $x'$ such that $H(x) = H(x')$ ).   |
| <b>Relationships &amp; Types (MDC vs. MAC)</b> | Hash functions are broadly classified based on key usage: 1. <b>Unkeyed Hash Function (Modification Detection Code, MDC)</b> : Used primarily to ensure <b>integrity</b> of the message; any alteration in the message will change the hash. Since no key is used, it does <b>not</b> provide authenticity. 2. <b>Keyed Hash Function (Message Authentication Code, MAC)</b> : Used to provide both <b>integrity</b> and <b>authentication</b> (confirming the source). <b>HMAC</b> (Hash-based MAC) is an efficient MAC derived from a cryptographic hash code like SHA-1. |
| <b>Examples &amp; Details</b>                  | <b>Unkeyed Examples:</b> MD5 and SHA-256. <b>Keyed Examples:</b> HMAC and CMAC. <b>SHA-512 Expansion:</b> This specific algorithm processes messages in large <b>1024-bit blocks</b> and produces a 512-bit digest. It involves 80 rounds of computation. <b>Collision:</b> Although a collision (where $H_1 = H_2$ and $M_1 \neq M_2$ ) can occur because of the many-to-one mapping, a scenario where $H_1 \neq H_2$ but $M_1 = M_2$ is impossible for a proper hash function, as it must be deterministic.   |
| <b>Revision Pointers (Key Takeaways)</b>       | Focus on the <b>three primary properties</b> (one-way, fixed-length output, collision resistance). Understand why hashing provides integrity but <i>not</i> confidentiality or authenticity (unless keyed). Remember the <b>many-to-one mapping</b> .   |
| <b>Practice &amp; Reflection Prompts</b>       | 1. If a secure hash function produces a 256-bit output, how many messages, in theory, would you need to hash to make a collision likely (Birthday Attack concept)? 2. Explain why MD5 or SHA-256 alone cannot provide non-repudiation, but they are essential building blocks for systems that do. 3. Define and differentiate between <b>first preimage resistance</b> and <b>second preimage resistance</b> .   |

---

## Topic 1.2: Digital Signatures and Certificates

| Section | Explanation |
|---------|-------------|
|---------|-------------|

|  |   |
|--|---|
| <b>Overview:<br/>Definition &amp; Concepts</b> | Digital signatures are the digital equivalent of handwritten signatures, designed to <b>bind pieces of digital data</b> with specific entities. They rely on <b>public-key technology</b> .   |
| <b>Theory: Key Properties</b>                  | Digital signatures provide three key security services: Integrity, Authentication (sender's identity confirmation), and <b>Non-repudiation</b> . Non-repudiation ensures a signer cannot later deny that they signed the message, as the signature can be verified using their public key.  |
| <b>Signing and Verification</b>                | <b>Signing:</b> The signer uses their <b>private key (\$d\$)</b> to generate the signature. An entity without knowledge of this private key cannot forge a valid signature. <b>Verification:</b> Anybody with access to the signer's <b>public key (\$e\$)</b> can verify the signature.  |
| <b>Relationships &amp; Types</b>               | Digital signatures often use hash functions: In a <b>signature with appendix</b> , a representative hash $H(M)$ is computed, and the signing transformation ( $f_s$ ) is applied to the hash, not the whole message.<br>**Types of Signatures:** 1. <b>Deterministic Signature:</b> Generates the <b>same signature</b> every time for a given message. 2. <b>Probabilistic Signature:</b> Uses randomization; generates different signatures even for the same message. 3. <b>Blind Signature:</b> The signer does not see the content of the message being signed. 4. <b>Undeniable Signature:</b> Requires the active participation of the signer during verification. |
| <b>Digital Certificates</b>                    | Digital certificates are needed to <b>bind public keys to entities</b> and establish the authenticity of public keys. They guard against malicious public keys. A certificate contains the entity's details (like name and address) and their public key, and it is <b>digitally signed by the private key of a trusted Certification Authority (CA)</b> . If a certificate is compromised or expires, the CA revokes it and maintains a <b>Certificate Revocation List (CRL)</b> .   |
| <b>Revision Pointers (Key Takeaways)</b>       | Ensure you know the difference between private key usage (signing) and public key usage (verification). <b>Non-repudiation</b> is the critical security service provided by digital signatures. Remember the role of the CA and CRL in managing digital certificates.   |
| <b>Practice &amp; Reflection Prompts</b>       | 1. Describe the steps involved in signature generation and verification for a "signature with appendix". 2. Why does a "Key-only attack" represent the most difficult type of attack to mount against a digital signature scheme? 3. Explain why deterministic signatures might offer less protection against certain forgery attacks compared to probabilistic signatures.   |

---

# Module 2: Secure Communication Protocols

## Topic 2.1: SSL/TLS and IPsec

| Section                                  | Explanation   |
|--|---|
| <b>Overview: SSL Objectives</b>          | <b>Secure Socket Layer (SSL)</b> was first used by Netscape to ensure the security of data sent via protocols like HTTP, LDAP, or POP3 over TCP/IP. It is primarily designed to provide reliable end-to-end secure service. The main objectives of SSL are: <b>Authentication</b> (of client and server), <b>Data Integrity</b> , and <b>Data Privacy (Confidentiality)</b> . <b>Faster Transmission</b> is NOT an objective.   |
| <b>Theory: SSL Architecture</b>          | SSL consists of two layers of protocols: 1. <b>SSL Record Protocol</b> : Responsible for data encryption and integrity, encapsulating application data into records (fragments of 16 Kbytes or smaller). 2. <b>Protocols for Connection Establishment</b> : Includes the Handshake Protocol (mutual authentication, key negotiation), ChangeCipherSpec Protocol, and Alert Protocol (for session messages and termination). <b>TLS (Transport Layer Security)</b> is an extension of SSL, aiming to provide security at the transport layer.  |
| <b>Relationship: S-HTTP</b>              | <b>Secure HTTP (S-HTTP)</b> is an extension of HTTP designed to send individual messages securely. It differs from SSL/HTTPS because SSL establishes a <b>secure connection</b> between two hosts, while S-HTTP secures <b>individual messages</b> . HTTPS works by running HTTP on top of SSL or TLS for secured transactions.   |
| <b>IP Security (IPSec)</b>               | IPSec builds security into the IP layer. It provides host-to-host or firewall-to-firewall encryption and authentication. IPSec provides protection through two distinct modes: 1. <b>Transport Mode</b> : Encapsulates <b>only the transport layer information</b> (the payload), leaving the original IP header intact. This is typically used between host nodes. 2. <b>Tunnel Mode</b> : Encapsulates the <b>entire IP packet</b> (header + payload) within IPSec protection. This mode allows tunneling between firewalls, hosts, or a mix of both. IPSec also offers Confidentiality (encryption), Authentication, Integrity, and <b>Replay Protection</b> . |
| <b>Revision Pointers (Key Takeaways)</b> | Know the <b>three core objectives of SSL</b> and the four main SSL protocols. Clearly differentiate between IPsec <b>Tunnel Mode</b> (entire packet protected) and <b>Transport Mode</b> (payload only protected).  |

|  |  |
|--|--|
| <b>Practice &amp; Reflection Prompts</b> | 1. If you are setting up a VPN between two corporate firewalls, which IPSec mode would be necessary and why? 2. Why is the ChangeCipherSpec Protocol message necessarily short, containing only the value of 1? 3. Explain the relationship between SSL and HTTPS, and how S-HTTP provides a different kind of security. |
|--|--|

---

## Module 3: Non-Cryptographic Hiding and Authentication

### Topic 3.1: Steganography and Digital Watermarking

| Section                                    | Explanation  |
|--|--|
| <b>Overview: Definition &amp; Concepts</b> | Steganography literally means “covered writing” (Greek). It is the <b>art of hiding the existence of a message</b> by embedding it inside another innocent medium (like audio, video, or image files). Unlike cryptography, where the message is observable but unintelligible, the goal of steganography is to prevent the detection of the secret message altogether.  |
| <b>Relationship: Digital Watermarking</b>  | Digital Watermarking is closely related to steganography, sharing the same functional behaviors. The key difference lies in the <b>intent of hiding</b> : Watermarking embeds information like copyright, ownership, or licensing.   |
| <b>Theory: Steganography in Images</b>     | Multimedia files (images, sound, movies) are commonly used as cover-media. Images are made up of pixels, where color is specified by components like Red, Green, and Blue, each typically represented by 8 bits. <b>LSB (Least Significant Bit) Insertion:</b> This is a common and simple method. It involves modifying the LSB of a pixel value based on the message to hide. Small changes in LSB values are generally unnoticeable by human observers.   |
| <b>Key Concepts &amp; Examples</b>         | <b>Compression Impact:</b> LSB insertion works well with lossless compression. However, it is <b>not robust against lossy compression like JPEG</b> because the LSB bits are easily lost during compression. <b>Capacity:</b> LSB steganography works better with 24-bit images because more bits per pixel result in a higher hiding capacity. For a 24-bit RGB image of size $200 \times 150$ , since each pixel has 3 channels (3 bits of hiding capacity), the total hidden bytes are $(200 \times 150 \times 3) / 8 = 11,250$ bytes. <b>Vulnerability:</b> LSB methods are vulnerable to image manipulation, filtering, or format conversion, which can destroy the hidden message. |

|  |   |
|--|---|
| <b>Revision Pointers (Key Takeaways)</b> | Understand the primary distinction between steganography and cryptography (hiding existence vs. hiding content). Know the properties and vulnerabilities of LSB insertion (simple, non-robust against lossy compression like JPEG, vulnerable to filtering).  |
| <b>Practice &amp; Reflection Prompts</b> | 1. Given an image size and bit depth, calculate the maximum steganographic capacity using 1-bit LSB insertion. 2. Why might steganography be used in conjunction with encryption, even though its primary goal is to avoid suspicion? 3. Compare LSB insertion with Masking and Filtering techniques for steganography. |

---

## Topic 3.2: Biometric Authentication

| Section                                  | Explanation   |
|--|---|
| <b>Overview: Definition</b>              | Biometrics is the <b>automated method for recognizing individuals</b> based on measurable biological (physiological) and behavioral characteristics.  |
| <b>Theory: Types of Biometrics</b>       | Biometrics are generally classified into two categories: 1. <b>Physiological Biometrics</b> : Based on physical body features, such as Fingerprint, Iris, Retina, Face, and Hand geometry. 2. <b>Behavioral Biometrics</b> : Based on patterns established by the individual, such as Signature dynamics, Keystroke patterns, and Gait.   |
| <b>Key Concepts &amp; Uniqueness</b>     | Different biometrics offer varying levels of uniqueness and invasiveness. <b>Iris and Retina recognition</b> provide the <b>highest uniqueness for identification</b> . However, retina recognition, while very secure, often requires effort from the subject and can be intrusive and stressful. Face, voice, and signature are generally less unique and more environment-dependent. |
| <b>Applications</b>                      | Biometrics are used widely in commercial applications (server login, electronic payment, access control) and government applications (passport/border control, Adhaar UID).   |
| <b>Revision Pointers (Key Takeaways)</b> | Be able to categorize biometrics as physiological or behavioral. Remember that Iris/Retina offer the highest uniqueness, but retina is invasive. Recognize that multimodal biometrics (e.g., Voice + Face) are often used in practical systems to improve accuracy.   |

|  |   |
|--|---|
| <b>Practice &amp; Reflection Prompts</b> | <p>1. Why are physiological biometrics generally considered more unique than behavioral biometrics? Give an example of each.</p> <p>2. Describe the trade-off between security and user experience when implementing retina recognition compared to fingerprint recognition.</p> <p>3. What is the difference between Authentication (1:1) and Verification (1:N) in the context of biometrics usage?</p> |
|--|---|

---

## Module 4: Network Attacks and Security

### Topic 4.1: Denial of Service (DoS) and Distributed DoS (DDoS)

| Section                             | Explanation   |
|-------------------------------------|---|
| <b>Overview: DoS Definition</b>     | A <b>Denial-of-Service (DoS) attack</b> is an explicit attempt by attackers to prevent legitimate users from accessing some service.  |
| <b>Relationship: DoS vs. DDoS</b>   | A <b>Distributed Denial-of-Service (DDoS) attack</b> differs because it utilizes <b>multiple compromised machines (a botnet)</b> to attack a single target. This coordination of many sources makes tracing and mitigation much harder than a single-source DoS attack.   |
| <b>Theory: Specific DoS Attacks</b> | <p><b>1. Smurf DoS Attack:</b> Sends <b>forged ICMP Echo Requests</b> to a network broadcast address, spoofing the source address to the victim's IP. Every host on the target network replies (ICMP Echo Reply) to the spoofed victim, flooding it.</p> <p><b>2. Ping-of-Death Attack:</b> Sends an ICMP echo packet that is <b>larger than the maximum allowed size (65,536 bytes)</b>. When the packet is reassembled, systems unable to handle the abnormality may crash or reboot.</p> <p><b>3. SYN Flooding Attack:</b> Exploits the TCP 3-way handshake. The attacker sends many connection requests (SYNC) with spoofed source addresses. The victim server allocates resources for each request and accumulates "half-open" connections until the queue is full, blocking legitimate requests.</p> |
| <b>HTTP Flood Attack (DDoS)</b>     | A type of DDoS designed to overwhelm a targeted server with HTTP requests.  |
|                                     | <p><b>a) HTTP GET Attack:</b> Bots coordinate to send multiple requests for assets (images, files) from the server until it is overwhelmed.</p> <p><b>b) HTTP POST Attack:</b> Exploits the fact that handling form submission (POST) and database persistence is resource-intensive compared to sending the request. Sending many POST requests quickly saturates server capacity.</p>   |

|  |   |
|--|---|
| <b>Defense Mechanisms</b>                | <p><b>Against SYN Flooding:</b> Use <b>cookies</b> to ensure the responder remains stateless until the initiator has produced at least two messages. <b>General Defense (ISPs):</b> Deploy <b>source address anti-spoof filters</b> (very important). Turn off directed broadcasts and develop traffic volume monitoring. <b>Against HTTP Floods:</b> Implement a computational <b>challenge</b> (like CAPTCHA or JavaScript test) to verify the requester is not a bot, or use a web application firewall.</p> |
| <b>Revision Pointers (Key Takeaways)</b> | <p>Master the distinction between DoS and DDoS (single vs. multiple sources). Know the mechanisms behind Smurf (broadcast/spoofing) and Ping-of-Death (oversized packet). Understand how SYN flooding exploits the TCP handshake.</p>   |
| <b>Practice &amp; Reflection Prompts</b> | <ol style="list-style-type: none"> <li>1. If an ISP implements anti-spoof filters, how would this impact the effectiveness of a Smurf attack?</li> <li>2. Describe the asymmetric state allocation problem that DoS attacks exploit, and how cookies mitigate this in SYN flooding.</li> <li>3. Identify two reasons why locating the source of a DDoS attack is challenging.</li> </ol>  |

---

## Topic 4.2: Domain Name System (DNS) Security

| Section                      | Explanation  |
|------------------------------|--|
| <b>Overview: DNS Purpose</b> | The <b>Domain Name System (DNS)</b> maintains the correspondence between human-readable <b>host names</b> (like <a href="http://www.google.com">www.google.com</a> ) and <b>machineusable IP addresses</b> . It stores this information in a hierarchical, distributed database.   |
| <b>Theory: DNS Services</b>  | DNS services include: <b>Host name to IP address translation</b> , host aliasing (many names for one host), and load distribution (a set of IP addresses for one canonical name). To resolve a query, the client typically contacts a root server, which directs it to the appropriate TLD server (e.g., <a href="http://.com">.com</a> ), which finally directs it to the authoritative server. |
| <b>Vulnerabilities</b>       | DNS queries and responses are often in plaintext, and <b>no authentication is done for DNS responses</b> , making it difficult to trust the source. Since DNS relies mostly on UDP packets, <b>IP address spoofing is relatively easy</b> .  |

|  |   |
|--|---|
| <b>DNS Cache Poisoning</b>               | <p><b>Basic Idea:</b> The goal is to give DNS servers false records and get them cached. The cache may be poisoned if a DNS server disregards the 16-bit request identifiers or accepts unsolicited DNS records.</p> <p><b>Procedure Summary:</b> An attacker sends a DNS query to an ISP's DNS server. Simultaneously, the attacker sends a forged DNS response, spoofing the authoritative server's IP address. If the forged response is accepted and cached, all downstream users of that ISP will be directed to the wrong, malicious website.</p> |
| <b>Defense: DNSSEC</b>                   | <p><b>DNSSEC</b> is a secure DNS server solution that guarantees the <b>Authenticity of DNS response origin, Integrity of DNS query response</b>, and authenticity of denial of existence. This is accomplished by using <b>public-key cryptography to digitally sign DNS responses</b> at every step.</p>  |
| <b>Revision Pointers (Key Takeaways)</b> | <p>Know the fundamental purpose of DNS (mapping names to IPs). Understand that the primary vulnerability exploited in poisoning is the lack of authentication/trust in DNS responses. DNSSEC uses digital signatures for response integrity.</p>  |
| <b>Practice &amp; Reflection Prompts</b> | <ol style="list-style-type: none"> <li>1. Why does DNS mostly relying on UDP packets make it easier for attackers to spoof IP addresses compared to using TCP?</li> <li>2. Describe the concept of DNS Caching and why top-level DNS servers are often cached.</li> <li>3. If a DNS cache poisoning attack is successful, what consequence would a regular user of that poisoned server face?</li> </ol>  |

---

### Topic 4.3: Other Network Security Topics (PGP, SSH, S-HTTP)

This section provides brief notes on related topics mentioned in the slides, often linked to application-layer security.

| Concept                          | Key Takeaways & Relationship  |
|----------------------------------|---|
| <b>PGP (Pretty Good Privacy)</b> | Provides confidentiality and authentication services for email and file storage. It is known for being free, based on well-known algorithms, and not controlled by governmental organizations. PGP uses five main services: Authentication, Confidentiality, Compression (using ZIP), E-mail Compatibility (using base-64 encoding), and Segmentation (max 50 KB segments). |
| <b>SSH (Secure Shell)</b>        | Originally developed as a secure replacement for protocols like Telnet and rlogin. It allows port forwarding (tunneling over SSH). SSH uses multiple authentication mechanisms, including straight passwords (protected by SSH encryption) and RSA-based authentication.  |

|               |  |
|---------------|--|
| <b>S-HTTP</b> | As noted in Topic 2.1, S-HTTP is an extension to the HTTP protocol that supports sending <b>individual messages securely</b> over the web. It differs from SSL/HTTPS, which establishes a secure connection. |
|---------------|--|

## Week 9 and 10

Greetings! I would be delighted to help you structure a comprehensive study and revision guide based on the materials provided. My goal is to deliver clear, insightful content that serves as an excellent reference for your final review.

This guide is organized logically, moving from external network and software exploits (Sniffing, Social Engineering, DoS) to internal hardware security topics (Hardware Attacks, Side-Channel Analysis, PUFs, and Trojans).

---

## Comprehensive Study and Revision Guide: Ethical Hacking

### I. Network and Software Exploits

#### Topic 1: Packet Sniffing and Network Analysis

| Section                          | Content Details & Concepts  |
|----------------------------------|---|
| <b>Overview &amp; Definition</b> | <b>Sniffing</b> (or network analysis) is the process of analyzing network activity by capturing network traffic. A sniffer is a program that monitors the data traveling across the network.  |
| <b>Key Concepts</b>              | <b>Promiscuous Mode:</b> When using sniffing tools like Wireshark, the Network Interface Card (NIC) is set to promiscuous mode. This allows the sniffer to <b>capture all packets</b> on the network segment, regardless of whether they are destined for the host machine. This usually requires root privileges. <b>Vulnerability:</b> Protocols that exchange data in plain text |

(unsecured form), such as **HTTP** and **FTP**, are vulnerable to sniffing attacks. Secure protocols like **HTTPS** and **SSL** are not vulnerable.

|  |   |
|--|---|
| <b>Examples &amp; Tools</b>              | <p><b>Wireshark:</b> An open-source network analyzer (or sniffer) tool. It captures, analyzes, and displays packets in a readable format and can generate statistical reports. <b>Wireshark cannot manipulate live traffic</b> or carry out attacks like SQL injection. <b>Ettercap:</b> A free, open-source tool primarily used for Man-in-the-Middle (MITM) attacks on LANs. Features include IP-based filtering, MAC-based filtering, <b>character injection</b> into established connections, and packet filtering/dropping. <b>BurpSuite:</b> An integrated platform for security testing of web applications. Its Proxy module allows users to <b>intercept, inspect, and modify raw traffic</b> between the user and the server.</p> |
| <b>Relationship to Other Topics</b>      | <p>Sniffing tools like Wireshark are passive analysis instruments, while tools like Ettercap and BurpSuite enable <i>active</i> attacks and traffic manipulation. Using encrypted protocols (like HTTPS) is a primary defense against sniffing, highlighting the weakness exploited by sniffers in unsecured communications (HTTP, FTP).</p>  |
| <b>Revision Pointers</b>                 | <p>Focus on the function of <b>promiscuous mode</b> and why switches are effective countermeasures (compared to hubs). Review which protocols are inherently vulnerable (HTTP, FTP). Understand the core distinction between Wireshark's capabilities (analysis only) and Ettercap/BurpSuite (traffic manipulation/MITM).</p>   |
| <b>Practice &amp; Reflection Prompts</b> | <p>1. If a LAN utilizes a switch, what limitation does this impose on a sniffer running on a single host, and why? 2. Explain the purpose of character injection in Ettercap during a live connection. 3. What types of sensitive information are protected when moving from Telnet to SSH, and how does this relate to packet sniffing countermeasures?</p>  |

---

## Topic 2: Social Engineering Attacks

| Section                          | Content Details & Concepts  |
|----------------------------------|---|
| <b>Overview &amp; Definition</b> | Social engineering is the <b>art of convincing people to reveal confidential information</b> . Attackers lure targets by exploiting their greediness (promising something for nothing). Attacks often target help desk personnel, technical support, and system administrators. |
| <b>Phases</b>                    | Social engineering attacks generally follow phases: 1. Research on the Target Company (dumpster diving, websites). 2. Select a Vulnerable Victim  |

(e.g., a greedy employee). 3. Develop a Relationship. 4. Exploit the Relationship to collect sensitive information.

|  |  |
|--|--|
| <b>Types &amp; Examples</b>              | <p><b>Human-based (Direct Interaction): Impersonation</b> (attacker pretends to be legitimate). <b>Shoulder Surfing</b> (using direct observation to gain info like passwords/PINs). <b>Piggybacking</b> (authorized person allows unauthorized person entry). <b>Tailgating</b> (unauthorized person follows authorized person using fake ID). <b>Reverse Social Engineering</b> (attacker acts as an authority, and the <i>victim approaches the attacker</i> for help, revealing sensitive information). <b>Computer-based: Phishing</b> (illegitimate email claims to be from a legitimate site to acquire personal data). Phishing is a computer-based attack, not human-based. Other examples include Pop-up Windows, Chain Letters, and Instant Chat Messengers. <b>Spear Phishing</b> is a direct, targeted phishing attack aimed at specific individuals.</p> |
| <b>Relationship to Other Topics</b>      | <p>Social engineering is an integral part of reconnaissance and access gaining, often preceding technical attacks. For example, information gathered via shoulder surfing (password, PIN) might enable an unauthorized network intrusion. The use of Trojans for defensive purposes (IC fingerprinting) requires trust in the manufacturing pipeline, which social engineering exploits if employees or vendors are targeted.</p>  |
| <b>Revision Pointers</b>                 | <p>Memorize the distinction between <b>human-based attacks</b> (Impersonation, Tailgating, Reverse Social Engineering) and <b>computer-based attacks</b> (Phishing). Understand the key difference between reverse social engineering and standard impersonation. Review physical security policies as key countermeasures.</p>  |
| <b>Practice &amp; Reflection Prompts</b> | <ol style="list-style-type: none"><li>1. Describe a scenario where an attacker might use reverse social engineering.</li><li>2. Why is two-factor authentication a strong defense against human-based social engineering attempts that successfully acquire a static password?</li><li>3. Categorize the following: Tailgating, Spear Phishing, Shoulder Surfing.</li></ol>  |

---

### Topic 3: Denial of Service (DoS/DDoS) Attacks

| Section                          | Content Details & Concepts   |
|----------------------------------|--|
| <b>Overview &amp; Definition</b> | <p>A <b>DoS attack</b> reduces, restricts, or prevents accessibility of system resources to legitimate users by flooding the victim system with non-legitimate service requests or traffic, overloading its resources. A <b>DDoS attack</b> involves a <b>multitude of compromised systems</b> (a botnet) attacking a single target.</p> |

|  |  |
|--|--|
| <b>Key Concepts</b>                      | <b>Botnet:</b> A huge network of compromised machines remotely controlled by an attacker, often used to launch DDoS attacks. <b>Goal:</b> To cause the unavailability of a website or slow network performance.  |
| <b>Techniques &amp; Examples</b>         | <b>SYN Flooding:</b> A classic attack that exploits a flaw in the TCP three-way handshake. The attacker sends many SYN requests with fake source IP addresses; the target sends SYN/ACK and keeps the partially opened connection in a small " <b>listen queue</b> " until it is exhausted, as the ACK is never received. <b>ICMP Flood Attack:</b> Sending a large number of ICMP packets to overwhelm the victim. <b>Application-Level Flood Attacks:</b> Exploiting programming weaknesses to prevent an application from processing legitimate requests. |
| <b>Tools</b>                             | <b>Slowloris:</b> Highly effective tool that works by opening thousands of connections to the targeted web server and holding them open by sending partial HTTP requests that are <i>never completed</i> . It requires minimal bandwidth. <b>LOIC (Low Orbit Ion Cannon):</b> Performs DoS by sending UDP, TCP, or HTTP garbage requests. <b>RUDY (R U Dead Yet?):</b> A "low and slow" attack designed to crash a server by submitting long form fields using an abnormally long <b>content-length</b> header.  |
| <b>Relationship to Other Topics</b>      | DoS and SQL injection attacks are explicitly defined as <b>software-based attacks</b> , contrasting with the later hardware-based attacks (Side-channel, Physical Probing). Botnets rely on previously compromised systems, linking back to general malware infection and possibly phishing/social engineering for initial access.   |
| <b>Revision Pointers</b>                 | Understand how <b>SYN Flooding</b> exploits the TCP handshake. Remember the key defining feature of Slowloris (partial requests, never completing them). Note that increasing bandwidth and replicating servers are critical physical countermeasures.   |
| <b>Practice &amp; Reflection Prompts</b> | 1. Why is a botnet necessary for launching a high-volume bandwidth attack? 2. Compare and contrast the approach of Slowloris versus SYN Flooding in exhausting server resources. 3. Briefly explain why DoS/DDoS attacks are categorized as software-based rather than hardware-based attacks.   |

---

## II. Hardware Security

### Topic 4: General Hardware Attacks & Countermeasures

| Section | Content Details & Concepts |
|---------|----------------------------|
|---------|----------------------------|

|  |  |
|--|--|
| <b>Overview &amp; Characterization</b>     | Hardware security focuses on protecting physical and digital components, including computer hardware (processors, firmware, memory) and mobile hardware (SIM Card, RFID, Smart Card, PUF). Attacks aim to exploit implementation vulnerabilities, even if cryptographic algorithms are mathematically secure.  |
| <b>Classification by Nature</b>            | <b>Invasive Attacks:</b> Attacks that require <b>direct physical access that alters or opens the chip/package</b> . These involve sophisticated instrumentation. <b>Non-Invasive Attacks:</b> Attacks that do not alter the chip; they typically measure external signals or test inputs/outputs.  |
| <b>Examples of Attacks</b>                 | <b>Physical Probing:</b> An <b>invasive</b> attack involving decapping the package or touching internal nodes with probes. <b>Reverse Engineering:</b> An <b>invasive</b> attack that requires physically exposing the circuit, removing layers, imaging dies, or destructively extracting the netlist. <b>Black-Box Testing:</b> A <b>non-invasive</b> attack that tests only inputs/outputs to infer the algorithm used. <b>Side-Channel Attack:</b> A <b>non-invasive</b> attack that measures external parameters like power, timing, or EM emissions.   |
| <b>Relationship to Other Topics</b>        | Side-channel attacks are a <i>type</i> of non-invasive hardware attack. Hardware Trojans are malicious modifications inserted during the design/fabrication pipeline, highlighting the supply chain vulnerabilities that these countermeasures try to address.   |
| <b>Key Takeaways &amp; Countermeasures</b> | Hardware countermeasures aim to prevent physical access or mask internal data leakage. Countermeasures include: <b>Obfuscating data</b> in registers and buses (scrambling/encrypting). <b>Adding dummy circuits</b> to generate random noise (specifically against side-channel attacks). <b>Adding metal mesh</b> on top of the circuit (if probed, it causes a short circuit and data reset). Using <b>Physical Unclonable Functions (PUFs)</b> for low-overhead security protocols. <i>Note: Using secure cryptographic algorithms is essential for data security but does not, on its own, mitigate hardware-based attacks.</i> |
| <b>Revision Pointers</b>                   | Clearly distinguish between invasive (Probing, Reverse Engineering) and non-invasive (Side-channel, Black-box) attacks. Focus on the countermeasures that rely on physical obstruction (metal mesh) and obfuscation (dummy circuits, data scrambling).   |

|  |   |
|--|---|
| <b>Practice &amp; Reflection Prompts</b> | 1. Why is increasing the CPU clock frequency generally ineffective (and potentially counterproductive) as a countermeasure against side-channel attacks? 2. If an attacker uses sophisticated measuring instruments to monitor delay during computation, which attack type are they executing? Is it invasive or non-invasive? 3. Explain how obfuscating the IC layout (e.g., using dummy circuitry) helps prevent hardware attacks. |
|--|---|

---

## Topic 5: Side-Channel Attacks (SCA)

| Section                                  | Content Details & Concepts  |
|--|---|
| <b>Overview &amp; Definition</b>         | SCA is a powerful, specific technique targeting the implementation (hardware or software) of cryptographic algorithms, not their mathematical security. It involves capturing <b>unintended leakage of information</b> during operation.  |
| <b>Vulnerabilities Exploited</b>         | Side channels exploited include <b>Timing</b> (time required for operations), <b>Power Consumption</b> , <b>Electromagnetic Emissions</b> , Heat, and Faulty Outputs.   |
| <b>Timing Analysis Attack</b>            | The execution time for a private key operation is dependent on the key in some way. For asymmetric key algorithms (like RSA or Diffie-Hellman), the time depends linearly on the number of '1' bits of the key (using the square-and-multiply algorithm). The calculation for modular exponentiation ( $x^{25}$ ) requires 4 squaring and 2 multiplication operations) demonstrates how computation steps relate directly to the binary representation of the exponent. |
| <b>Power Analysis Attack</b>             | A more effective form of SCA that analyzes the power consumed by a device during a cryptographic operation to extract key information.  |
| <b>Simple Power Analysis (SPA)</b>       | The attacker visually examines power consumption waveforms to learn bits of the secret key. SPA can identify <b>big features</b> like rounds of DES/AES or the difference between square vs. multiply operations in RSA exponentiation.   |
| <b>Differential Power Analysis (DPA)</b> | <b>More sophisticated and effective</b> than SPA. DPA requires <b>multiple measurements</b> . It involves partitioning the data and related power curves based on selected bits, taking the difference, and looking for correlation peaks that correspond to a correct key guess.   |

|  |   |
|--|---|
| <b>Countermeasures</b>                   | For timing analysis, the primary countermeasure is to <b>make the execution time data independent</b> (e.g., modifying algorithms so the time is $n \cdot (t_{\text{square}} + t_{\text{mul}})$ ). For power analysis, common approaches include: hardware masking schemes, adding random noise or noise generators, and software techniques like introducing redundant computations or using internal randomization (adding/multiplying a random value to intermediate results) to mask data representation. |
| <b>Revision Pointers</b>                 | Be able to compare SPA (visual examination, single measurement features) vs. DPA (multiple measurements, difference curve analysis, more effective). Understand why cryptographic algorithms alone are insufficient defense against SCA—it's an implementation issue.   |
| <b>Practice &amp; Reflection Prompts</b> | 1. Given the binary representation of the exponent $N$ , explain how modular exponentiation reveals information about the secret key in a timing analysis attack. 2. Why does DPA require multiple measurements, unlike SPA? 3. How does adding a random noise generator mitigate side-channel attacks?   |

---

## Topic 6: Physical Unclonable Functions (PUFs)

| Section                          | Content Details & Concepts  |
|----------------------------------|---|
| <b>Overview &amp; Definition</b> | A PUF is the " <b>fingerprint</b> " of some device. It is a challenge-response mechanism where the output ("response") depends on the unpredictable, instance-specific variations inherent to the physical material and manufacturing process of the IC.  |
| <b>Main Purpose</b>              | The main purpose of PUF in hardware security is to provide <b>device-unique authentication</b> and secure key generation. PUFs exploit manufacturing variations, which are usually a challenge in design, for security benefits.  |
| <b>Desirable Properties</b>      | <b>Evaluatable:</b> Given the PUF and the input challenge ( $x$ ), it must be <b>easy to evaluate</b> the response ( $y = \text{PUF}(x)$ ). <b>Unclonable:</b> It is hard to construct an imitation PUF' that matches the original PUF's responses for all challenges ( $\text{PUF}' \neq \text{PUF}$ , but $\text{PUF}'(x) = \text{PUF}(x)$ for all $x$ ). <b>One-way:</b> Given only the response ( $y$ ), it is hard to find the original challenge ( $x$ ). |

|  |   |
|--|---|
| <b>Advantages &amp; Applications</b>     | PUFs reduce cost and increase security. The intrinsic properties of the device generate the secret key; the key <b>never leaves the IC's cryptographic boundary</b> and is not stored in Non-Volatile Memory (NVM). Applications include <b>device identification</b> (protecting against counterfeits/ASIC substitution) and <b>Private/Public Key Pair Generation</b> (PUF response acts as a random seed).   |
| <b>Practical Designs</b>                 | <p><b>Arbiter PUF:</b> Composed of switching stages; the challenge selects a path, and an arbiter compares the accumulated delay to give a 1-bit decision.</p> <p><b>Ring Oscillator PUF (ROPUF):</b> The challenge selects two Ring Oscillators (ROs); process variation ensures they have different oscillation frequencies, which are compared.</p> <p><b>SRAM PUF:</b> Uses the power-up initial value of an SRAM cell as the response, based on slight mismatches in driving capabilities/routing delays in the silicon space.</p> |
| <b>Revision Pointers</b>                 | Ensure you know the definitions of the three desirable properties (Evaluable, Unclonable, One-way) and which statement describes each. Understand the core benefit: keys are generated intrinsically and not stored externally in NVM.  |
| <b>Practice &amp; Reflection Prompts</b> | 1. Explain the role of manufacturing process variations in PUF design. 2. What is the benefit of using a PUF response as the seed for a private/public key generation algorithm? 3. Why is the Evaluability property of a PUF considered desirable?   |

## Topic 7: Hardware Trojans (HT)

| Section                          | Content Details & Concepts  |
|----------------------------------|---|
| <b>Overview &amp; Definition</b> | A <b>Hardware Trojan (HT)</b> is a <b>malicious logic modification</b> (change) inserted into a chip's circuitry. This modification happens without the knowledge of the designer or user, typically during the design or fabrication stages. Trojans are inserted at the <b>hardware level</b> . |
| <b>Mechanism</b>                 | A HT is characterized by its physical representation and its behavior. It consists of a <b>trigger condition</b> and a potentially malicious <b>payload</b> . They have extremely small hardware overhead, making them stealthy.  |
| <b>Context &amp; Insertion</b>   | The rise of fabless manufacturing, reliance on third-party IPs, and outsourcing fabrication to offshore fabs increases the opportunities for untrusted parties to insert Trojans.   |

|  |  |
|--|--|
| <b>Detection Challenges</b>              | Trojan detection is <b>difficult</b> . Key challenges include: high overhead, stealthiness, very low observability of trigger nodes, and the extremely large number of Trojan types possible. Crucially, <b>no single method can detect all types of hardware Trojans</b> . The detection process often involves significant design, testing, or runtime overhead. |
| <b>False Assumptions</b>                 | It is <b>false</b> that detection of hardware Trojans is relatively easy. It is also <b>false</b> that Hardware Trojans are always inserted at the software level; they are inserted at the hardware level.  |
| <b>Relationship to Other Topics</b>      | HTs represent the ultimate threat in hardware security because they are difficult to detect and leverage the compromised supply chain. PUFs can sometimes be used in defensive protocols to make hardware-based attacks, like side-channel or Trojan detection, more difficult.  |
| <b>Revision Pointers</b>                 | Understand the definition: <i>malicious modification inside a chip</i> . Focus on the components (trigger + payload). Review the reasons why detection is so challenging (stealth, lack of a universal detection method).  |
| <b>Practice &amp; Reflection Prompts</b> | 1. Why does the prevalence of a fabless manufacturing model contribute to the risk of hardware Trojan insertion? 2. Describe the two primary components that characterize a Hardware Trojan. 3. If a security feature caused unauthorized changes in a circuit to be detected, how might that relate to the defensive use of Trojans?                              |

Week 11 and week 12

A truly comprehensive study guide is an invaluable asset! Drawing upon the assignment solutions and the lecture slide material, I have created a detailed revision resource, organized for clarity and maximum study effectiveness.

---

## Comprehensive Study and Revision Guide: Ethical Hacking Tools and Techniques

# Module 1: The Metasploit Framework (Exploitation and Validation)

## Overview

Metasploit is an **open-source penetration testing platform** that is highly useful for security auditing, containing information-gathering tools, modules, and an exploit development environment. It is available in operating systems like Kali Linux and can be installed on Windows/Linux/MacOS.

## Key Concepts and Modules

The Metasploit Framework organizes its components into specific modules:

| Module           | Definition and Purpose  | Key Concepts / Clarifications   |
|------------------|---|---|
| <b>Exploit</b>   | A piece of code designed to <b>take advantage of system or application bugs (vulnerabilities)</b> . It is the basic module used for intrusion.  | The framework offers over 2000 exploits for various operating systems (Windows, Linux, Android, Mac).   |
| <b>Payload</b>   | Consists of malicious codes (over 500 available) used to <b>establish a communication channel</b> between the Metasploit framework and the target system.                               | <b>Meterpreter</b> is a powerful, interactive payload that allows remote interactive access, VNC screen control, file transfers (browse, upload, download), and privilege escalation support. Other types include Command Shell (runs scripts/commands), Dynamic payloads (evade anti-virus by generating unique code), and Static payloads (enable static IP/port forwarding). |
| <b>Auxiliary</b> | An additional module used for performing non-exploitive tasks such as <b>brute force attacks, Denial of Service (DoS) attacks, host and port scanning, and vulnerability scanning</b> . | It cannot grant control like exploits or payloads, but it is powerful for reconnaissance.   |

|                            |   |  |
|----------------------------|---|--|
| <b>Encoder</b>             | Used to <b>encode payloads to evade anti-virus detection.</b><br>Anti-virus software often searches for bad hexadecimal codes, which the Encoder module helps bypass. | Over 45 encoding schemes are available.  |
| <b>NOPS (No Operation)</b> | Helps <b>prevent the payload from crashing</b> by generating no-operation instructions if the payload is blocked.   | Provides additional support to payloads. |
| <b>POST</b>                | Used to perform <b>deeper penetration testing</b> after the attacker has already gained initial access to the target system.  | Post-exploitation activities.            |

## Relationships and Workflow

The core interaction follows a sequence:

1. **Scanning/Discovery:** Find a vulnerability (using NMAP or Metasploit Auxiliaries).
2. **Selection:** Pick an **Exploit** based on the vulnerability.
3. **Configuration:** Configure the exploit (set target IP/port).
4. **Payload Selection:** Pick a **Payload** (e.g., Meterpreter or Command Shell).
5. **Evasion:** **Encode** the payload (using the Encoder module).
6. **Execution:** Execute the **Exploit** to deliver the payload.

## Key Configurations (RHOST/RPORT vs. LHOST/LPORT)

To configure an exploit, specific parameters must be set:

- **RHOST:** Target (remote) machine IP address.
- **RPORT:** Target (remote) port number.
- **LHOST:** Attacker (local) machine IP address.
- **LPORT:** Local port on the attacker machine.

## Revision Pointers

- **Focus on Module Differentiation:** Be able to instantly distinguish the purpose of the Exploit (gaining access) vs. Payload (establishing communication) vs. Encoder (evasion) vs. Auxiliary (scanning/brute force).
- **Meterpreter:** Understand why Meterpreter is considered a powerful, interactive payload.

- **Configuration:** Memorise the meaning of RHOST, RPORT, LHOST, and LPORT.

## Practice and Reflection Prompts

1. If a target system is running vulnerable web application software, which Metasploit module would you use first to confirm the vulnerability before launching an attack? Why?
  2. Describe the function of the Encoder module and explain how it helps dynamic payloads successfully penetrate a target protected by anti-virus software.
  3. You are setting up a reverse TCP connection. Which options must you set for the attacker's machine, and which must you set for the target?.
- 

# Module 2: Web Application Attacks (SQL Injection & XSS)

## Topic A: SQL Injection (SQLI)

### Definition and Concepts

SQL injection is a technique that exploits **non-validated input vulnerabilities** to pass harmful SQL commands through a web application for execution by a backend database. These attacks target websites that fail to follow secure coding practices.

**Impacts of SQLI** include information disclosure, reputation decline, compromised data integrity/availability, and denial of service (DoS).

### Types of SQL Injection

1. **Error-Based SQL Injection:** Attackers insert bad input to cause the application to throw database errors, which are then analyzed to find vulnerabilities.
2. **Union Query-Based SQL Injection:** Combines a valid SQL query with an invalid one.
3. **Blind SQL Injection:** Used when the application shows generic errors or no output (it does not print anything for incorrect user input). The attacker determines the answer based on the application's response (e.g., timing or slight differences in content).
  - *Boolean-based blind:* Analysing query output character by character.
  - *Time-based blind:* Analysing query output based on the time taken for execution.
4. *Note:* **Opcode injection** is *not* a type of SQL injection technique.

### SQLMAP Tool and Commands

**SQLMAP** is an **open-source penetration testing tool** that automates the detection and exploitation of SQL injection flaws. It supports almost all types of databases (MySQL, Oracle, Microsoft SQL Server, etc.) and all SQL injection techniques.

### Key SQLMAP Commands:

| Command        | Purpose  | Assignment Confirmation                |
|----------------|--|--|
| --dbs          | Lists <b>all database names</b> available on the target.         | Correct option to list database names. |
| --tables       | Lists tables within a specified database (-D).                   | Lists tables.                          |
| --dump         | Dumps database table entries (retrieves the data).               | Dumps entries.                         |
| --current-db   | Detects the current database being used.                         | Shows current DB.                      |
| --users        | Retrieves database user names (credential-related).              | Valid option to extract credentials.   |
| --passwords    | Retrieves user passwords, typically hashed (credential-related). | Valid option to extract credentials.   |
| --current-user | Recover the session user (credential-related).                   | Valid option to extract credentials.   |
| --hostname     | Gets the DBMS server name.                                       | Returns DBMS host, not credentials.    |

## Topic B: Cross-Site Scripting (XSS)

### Definition and Concepts

Cross-Site Scripting (XSS) is a type of **injection attack** where malicious scripts (generally browser-side scripts) are injected into websites and sent to a different end user.

The attack works because the end user's browser executes the script, thinking it came from a trusted source. This allows the malicious script to access sensitive information like cookies, session tokens, or even rewrite the HTML content. XSS flaws occur wherever a web application uses user input in its output without proper validation or encoding.

### Types of XSS Attacks

The three classical forms of XSS are Stored, Reflected, and DOM-based.

#### 1. **Stored XSS (Persistent or Type I):**

- Occurs when malicious user input is **stored on the target server** (e.g., in a database, message forum).
- When a victim retrieves the stored data, the malicious code executes in their browser.

- *Example:* An attacker submits malicious code via a feedback form; when the admin opens the feedback, the payload executes.
- 2. **Reflected XSS (Non-Persistent or Type II):**
  - Occurs when user input is **immediately returned** by the web application (e.g., in an error message or search result).
  - Attacks are often delivered via external approaches, such as email, where clicking a link delivers the payload.
- 3. **DOM-Based XSS (Type 0):**
  - The entire malicious data flow, from source (where data is read, e.g., URL) to sink (the sensitive method call that executes the data, e.g., `document.write`), happens **within the browser** (Document Object Model). The data flow never leaves the browser.

#### **Advanced Classifications (Server vs. Client XSS):**

- **Server XSS:** Occurs when untrusted user data is included in an HTTP response generated by the server. This can be Reflected Server XSS or Stored Server XSS. The vulnerability is entirely in the server-side code.
- **Client XSS:** Occurs when untrusted user data is used to update the DOM using an unsafe JavaScript call. This can be Reflected Client XSS or Stored Client XSS.

#### **Revision Pointers**

- **SQLI Types vs. Output:** Understand that Error-based produces output (errors), while Blind SQLI produces no output for incorrect input.
- **SQLMAP Functions:** Focus on the specific commands used to list databases (`--dbs`) and extract credentials (`--users`, `--passwords`).
- **XSS Persistence:** Distinguish between Stored (persistent data on the server) and Reflected (non-persistent, immediately bounced back) XSS.

#### **Practice and Reflection Prompts**

1. Explain the key difference between Error-based SQL Injection and Blind SQL Injection. Under what circumstances would an attacker rely solely on the blind technique?.
  2. If you are using SQLMAP and successfully identify a vulnerability, list the SQLMAP commands necessary to retrieve the entire list of databases, the tables within one database, and finally dump the entries of a specific table.
  3. A malicious script is injected into a message board post and executes every time a user views the post. Which XSS type is this, and why is it considered more dangerous than reflected XSS?.
- 

## **Module 3: NMAP (Network Mapper)**

## Overview

NMAP is a **free, open-source tool** used for **vulnerability scanning and network discovery**. Network administrators use it to discover active hosts and services, find open ports, detect security risks, and determine OS versions.

**NMAP's Main Features:** Host Discovery, Port Scanning, Service and Version Detection, and OS Detection.

### A. Host Discovery Techniques

Host discovery, or ping sweep, is the basic step in network mapping to determine which hosts are alive. The key is sending probes and analyzing the replies.

| Technique           | Command Option   | Concept / Reply Indicates Alive Host  |
|---------------------|--|---|
| ICMP Echo Sweep     | <code>-PE</code>   | Sends an ICMP Echo Request (Type 8). <b>ICMP Echo Reply (Type 0) indicates the target is alive.</b> |
| Non-Echo ICMP       | <code>-PP</code> (Timestamp Request, Type 13); <code>-PM</code> (Address Mask Request, Type 17). | Uses other ICMP messages; the target responds to these.   |
| TCP Sweep           | <code>-PS</code> (TCP SYN sweep); <code>-PA</code> (TCP ACK sweep).                              | Sends TCP SYN or ACK packets to target ports (e.g., 21, 80).  |
| UDP Sweep           | <code>-PU</code> or <code>-sU</code> .   | Sends a UDP datagram. <b>No ICMP PORT UNREACHABLE message indicates the target is alive.</b>        |
| Skip Host Discovery | <code>-Pn</code>   | <b>Treats all hosts as online</b> (useful when a firewall is blocking ping probes).                 |

### B. Port Scanning Techniques

Port scanning determines which services are running (listening on TCP or UDP ports). By default, **NMAP scans the top 1000 ports**.

| Technique                             | Concept / Workflow  | Key Characteristic                                  |
|---------------------------------------|---|---|
| TCP Connect Scan ( <code>-sT</code> ) | Uses the <b>complete 3-way handshake</b> (SYN, SYN/ACK, ACK). | Easy to detect as it establishes a full connection, |

|                                 |   |  |
|---------------------------------|---|--|
|                                 |   | which is logged by the system.   |
| <b>TCP SYN Scan (Half-open)</b> | Sends SYN, receives SYN/ACK, but then immediately terminates with RST <b>without completing the 3-way handshake.</b>  | Considered "half-open scanning" or stealthier than Connect scan.                 |
| <b>TCP Stealth Scan</b>         | Aims to avoid detection and logging by using specific TCP flags (e.g., FIN, Null, Xmas probes). <b>Closed ports reply with RST; open ports ignore the packet.</b> | Slow scan rate, relying on inverse mapping (response only sent by closed ports). |
| <b>FTP Bounce Scan</b>          | Connects to an FTP server and asks the server to initiate an active data transfer process to the actual target.   | Quite slow.  |

#### Port Selection Commands:

- `-p <port ranges>`: Scan only specified ports (e.g., `-p22`).
- `--top-ports <number>`: Scans the specified number of **most common ports** (e.g., `--top-ports 10`).
- `-F`: Fast mode, scans fewer ports than the default 1000.

## C. Service, Version, and OS Detection

Once ports are found open, NMAP can probe them for detailed information.

| Detection Type                   | Command Option                            | Concept  |
|----------------------------------|---|--|
| <b>OS Detection</b>              | <code>-O</code>                           | Enables OS detection by analyzing how the target's IP stack implementation responds to various probes (e.g., FIN probe, TCP initial sequence number sampling). |
| <b>Service/Version Detection</b> | <code>-sV</code>                          | Probes open ports to detect the <b>application name and version number</b> .   |
| <b>Scripts</b>                   | <code>--script &lt;script name&gt;</code> | NMAP has thousands of scripts (NSE - Nmap Scripting Engine) to perform complex operations, like checking vulnerability status ( <code>--script vuln</code> ).  |

## Relationships

- **Host Discovery is Primary:** Host discovery (-PE, -Pn) must occur before or alongside port scanning to ensure the target is online.
- **Scanning vs. Detection:** Port scanning (-sT, -sS) determines *if* a service is listening. Service/Version/OS detection (-sV, -O) determines *what* specific software or OS is running on those open ports.

## Revision Pointers

- **TCP Scan Difference:** Understand the fundamental difference between Connect scan (full handshake, loud) and SYN scan (half-open, stealthier).
- **Core NMAP Commands:** Memorise the options for essential tasks: Host Discovery (-PE), Port Scanning Default/Connect (-sT), Skipping Host Discovery (-Pn), OS Detection (-O), and Version Detection (-sV).
- **Port Defaults:** Remember that NMAP defaults to scanning the top 1000 ports.

## Practice and Reflection Prompts

1. A firewall is configured to drop all incoming ICMP Echo Requests (Type 8). Which NMAP host discovery option should you use to ensure the scan treats the hosts as online, and why is this command necessary?
  2. If you run a TCP Connect Scan (-sT), describe the exact packet exchange (the 3-way handshake) that confirms the port is OPEN. How does this differ from the exchange if the port is CLOSED?
  3. You have successfully identified an open port (80/TCP). Which NMAP option would you use next to find out if the service is running Apache HTTP Server version 2.4.41?.
- 

## Module 4: Wireshark (Network Analysis)

### Overview

Wireshark is an **open-source network protocol analyzer** (or "sniffer") used for profiling network traffic and analyzing packets. It captures network data and displays it in a readable format, often used for logging traffic for forensics or analysis.

### Key Concept: Promiscuous Mode

For Wireshark to function effectively, the machine's Network Interface Card (NIC) is typically put into **promiscuous mode**.

- In this mode, the NIC captures **all packets in the same network segment** to which the host is connected, not just packets specifically oriented to that host.

- Setting the NIC to promiscuous mode usually requires root privileges.

## Default Capture Format

Wireshark saves captured packets by default in the **.pcapng (Packet Capture Next Generation)** format (or .pcap in older versions). It can export reports to other formats like TXT, CSV, or XML.

## Analysis and Filtering

Wireshark provides various menus for analysis:

- **Applying Filters:** Filters are used to restrict the packets displayed in the summary window (e.g., TCP filter, IP filter). The filter bar turns green for a correct filter and red for a wrong filter.
- **Follow Stream:** The **Follow** option (under the Analyze menu) allows the user to see the complete detail of the packets in a conversation (e.g., HTTP/TCP stream). This is often used to view sensitive information like plain-text login credentials from unsecured HTTP websites.
- **Statistics Menu:** Used to check statistics about the capturing, such as protocol hierarchy, number of packets sent/received, and connection details.

## Relationships

- **NMAP and Wireshark:** NMAP generates network traffic (probes, scans) to discover information. Wireshark can be run simultaneously on the network segment to observe and analyze the exact packets NMAP is sending and receiving.

## Revision Pointers

- **Promiscuous Mode:** Crucially, understand what promiscuous mode enables the NIC to capture (all segment traffic, not just its own).
- **Default Output:** Know the default file format for captured packets (.pcapng/.pcap).
- **Analysis Function:** Be aware that the **Follow** stream option is often used to quickly extract application-layer data (like credentials) from conversations.

## Practice and Reflection Prompts

1. Explain why placing a NIC into promiscuous mode is essential for comprehensive network sniffing, and why an attacker cannot capture traffic from entirely different network segments if a switch is used.
2. After performing a capture using Wireshark, what is the default file extension, and which menu options would you use if you needed to export the packet data specifically for a security report in CSV format?.

3. If you capture a login attempt on an unencrypted HTTP website, describe the Wireshark analysis steps you would take to immediately view the username and password used in plain text.