**LECTURE No. 1**

**Introduction to Ethical Hacking**

# Ethical Hacking

* Ethical Hacking is a process of detecting or locating vulnerabilities (weaknesses) in an application, system, or organization's infrastructure that an attacker can use to exploit an individual or organization.
* It is also known as ***penetration testing***, ***intrusion testing***, or ***red teaming***.

How many ways can you enter into the system? It is checked by an ethical hacker.

**Question: -** *What is the difference between vulnerabilities and weaknesses*?

# Ethical Hacking Terminologies

Here are some common terminologies associated with ethical hacking:

## Ethical Hackers

1. Employed by companies to perform penetration tests.
2. These are the known people of the companies.
3. Their job is to search for vulnerabilities in the systems.

Solutions can also be performed but extra charges would be bear.

## Penetration Test

1. Legal attempt to break into the company’s network to find the

weak links.

1. Tester only reports findings and does not provide solutions.

## Security Test

1. Also includes analyzing company security policy and procedures.
2. Tester offers solutions to secure or protect the network.

## Hacking

1. Hacking is a process.
2. Showing computer expertise.
3. This means he/she (hacker) is the more intelligent person in the network than the person who is trying to secure the network.

## Cracking

1. Breaching/breaking security on systems or software.
2. Unlike ethical hacking, which is performed to improve security, cracking is often associated with illegal activities

## Spoofing

1. A scammer pretends to be someone else.
2. Faking the originating IP address in a datagram. (This point will be easily understandable in future lectures; means to say ***what is IP Address in the datagram***)

## Denial of Service (DoS)

1. A hacker floods a website or online service with so much traffic that it crashes or becomes unavailable to users.
2. It's like a bunch of people crowding the entrance to a store so no one else can get in.

## Port Scanning

1. Port scanning is like knocking on different doors of a house to see which ones are open.
2. In computer terms, it's a technique used to find out which ports on a computer or network device are open and listening for connections.

# Gaining Access to the System

There are different ways in which you can gain access to a system. Here are some common ways are exploring...

## Front Door

1. Password Guessing
2. Password/Key Stealing

## Back Door

1. A backdoor is a hidden method for bypassing security controls to gain access to a system or network. It is often installed by attackers to maintain persistent access.
2. Often left by original developers as debugging and diagnostic tools.

## Trojan Horses

1. Usually hidden inside the software. The software which we download and install from the net.
2. These are normally installed back doors in the system.

## Software Vulnerability Exploitation

1. This is a playground for those hackers, who are hacking first time.
2. When software is made and bought by the company or individual, by the time, the user or company knows about the vulnerabilities of that specific software, then the software house publishes these vulnerabilities and patches that remove these vulnerabilities***. If users don’t use these patches, then it is a golden opportunity for the hackers.***

# Once Inside, The Hacker Can….

1. Modify Logs (Logs which have the information about the login details of the users, Hackers modify these logs so that when the administrator/security professionals come and see the logs, he/she cannot judge that his/her system/network has been hacked.)
2. Steal Files
3. Modify Files
4. Install Back Doors
5. Attack Other Systems (Hackers can attack another system in the network from your system.)

# The Role of Security and Penetration Testers

## Script Kiddies or Packet Monkeys

1. Young and inexperienced hackers
2. Copy Code and techniques from knowledgeable hackers.

## Experienced Penetration Testers

1. Experienced Hackers
2. Write programs or scripts using programming languages like C, C++, Python, JavaScript, SQL, etc.

# Penetration-Testing Methodologies

Here are some common methodologies associated with the Penetration-Testing:

## Tiger Box

1. The tester uses a pre-configured set of tools (often referred to as a "Tiger Box") to test the security of a system.
2. Collection of OSs and hacking tools.
3. Usually on a Laptop
4. Helps penetration testers and security testers conduct vulnerability assessments and attacks.
5. This term/methodology is less commonly used.

## White Box Model

1. The tester is told everything about the network topology and technology
2. The tester is authorized to interview IT Personnel and company employees
3. Makes tester’s job a little easier.

## Black Box Model

1. The tester is not given details about the network
2. The burden is on the tester to find the details

## Gray Box Model

1. Hybrid of the white and black box models
2. The company gives partial information to a tester.

**LECTURE No. 2**

**Basic Concepts of Networking**

# Computer Network

A communication system for connecting computers/hosts (devices with connectivity/communication property).

# Why We Need Computer Network?

* Better Connectivity
* Better Communication
* Better Sharing of Resources
* Better People Together

# Types of Computer Networks

Broadly Computer Networks are divided into two major types that are:

1. Local Area Network (LAN)
2. Wide Area Network (WAN)

## Local Area Network

1. It is faster
2. It is cheaper in the long run
3. Connects hosts within a relatively small geographical area like

* Same Room
* Same Building
* Same Campus

## Wide Area Network

1. It is slower.
2. It is Expensive in the long run.
3. Hosts may be widely dispersed like

* Across Campuses
* Across Cities/Countries/Continents

# Data Communication over Network

For communicating the data over the network, we normally used two approaches like

1. Circuit Switching
2. Packet Switching

## 

**Figure 1:** Basic Network

## Circuit Switching

Circuit switching is a method of communication where a dedicated communication path or circuit is established between two endpoints for the duration of the communication session.

### Key Characteristics

Dedicated Path: A dedicated communication path is reserved between the sender and receiver.

Fixed Bandwidth: The entire bandwidth of the circuit is reserved for the duration of the session, even if no data is being transmitted.

Continuous Transmission: Data is sent as a continuous stream of bits once the circuit is established.

Delay: Initial setup delay to establish the circuit, but minimal delay once the circuit is established.

Resources: Inefficient resource utilization as the dedicated circuit cannot be used by others until the session is terminated.

### Process

Connection Establishment: A connection setup phase where a dedicated path is established between the sender and receiver through a series of switches.

Data Transfer: Data is transmitted continuously over the established path.

Connection Teardown: The connection is terminated, and the dedicated resources are released.

### Advantages

* Consistent and reliable communication with a guaranteed path and bandwidth.
* Minimal transmission delay once the circuit is established.
* Useful for voice communication.

### Disadvantages

* Inefficient use of resources as the dedicated path remains reserved even during idle periods.
* High initial setup time to establish the circuit.

### Example

Traditional telephone networks, where a dedicated line is established for the duration of the call.

## Packet Switching

Packet switching is a method of communication where data is divided into smaller packets, each of which can take different paths to reach the destination. This method is widely used in modern computer networks, including the Internet.

### Key Characteristics

No Dedicated Path: No dedicated communication path; packets are routed independently.

Dynamic Bandwidth: Bandwidth is dynamically allocated and shared among multiple users.

Discontinuous Transmission: Data is sent in small packets, which may travel different routes and arrive out of order.

Delay: Variable delay due to packet routing and possible congestion.

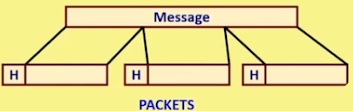
Resources: Efficient resource utilization as the network can handle multiple communications simultaneously.

### Process

Data Segmentation: Data is divided into small packets, each containing a header with routing information.

Packet Routing: Packets are routed independently through the network, using the most efficient path at any given time.

Reassembly: Packets are reassembled into the original data at the destination.



**Figure 2:** Packet Switching

### Advantages

* Efficient use of network resources as multiple communications share the same network paths.
* Robustness and fault tolerance, as packets can be rerouted in case of network failures.

### Disadvantages

* Variable transmission delay and potential packet loss due to network congestion.
* Complexity in packet reassembly and error handling.

### Example

The Internet and most modern computer networks, where data is transmitted in packets.

# Comparison

**Table 1:** Comparison between Circuit Switching and Packet Switching

|  |  |  |
| --- | --- | --- |
| Feature | Circuit Switching | Packet Switching |
| Path | Dedicated path | No dedicated path |
| Bandwidth | Fixed bandwidth | Dynamic bandwidth |
| Data Transmission | Continuous | Discontinuous |
| Initial Delay | High (connection setup) | Low (no connection setup) |
| Transmission Delay | Low (once circuit established) | Variable (due to routing and congestion) |
| Resource Utilization | Inefficient (reserved resources) | Efficient (shared resources) |
| Example | Traditional telephone networks | Internet and modern computer networks |

# Packet Switching Approaches

Packet switching can be further classified into two main approaches: virtual circuit switching and datagram switching. Here's a detailed explanation of each:

## Virtual Circuit Switching

Virtual circuit switching is a type of packet switching where a logical connection, called a virtual circuit, is established between the sender and receiver before any packets are transmitted. Although the data is sent in packets, each packet follows the same path through the network.

### Key Characteristics

Pre-established Path: A virtual path is established before data transmission begins.

Fixed Route: All packets follow the same route through the network.

Sequence Preservation: Packets arrive in the order they were sent.

Connection Setup: Requires an initial setup phase to establish the virtual circuit.

### Process

Connection Establishment: A virtual circuit is established through the network. This involves negotiating a path and setting up routing information in the network switches/routers.

Data Transfer: Packets are transmitted along the established virtual circuit. Each packet contains a Virtual Circuit Identifier (VCI) rather than the full destination address.

Connection Teardown: The virtual circuit is terminated after the data transfer is complete, and resources are released.

### Advantages

* Reliable and ordered delivery of packets.
* Simplified routing decisions as all packets follow the same path.

### Disadvantages

* Initial setup time to establish the virtual circuit.
* Less flexible in handling dynamic changes in the network.

### Example

Asynchronous Transfer Mode (ATM) and Frame Relay networks use virtual circuit switching.

## Datagram Switching

Datagram switching, often called datagram or connectionless packet switching, is a type of packet switching where each packet is treated independently and routed separately. Each packet contains the full destination address and may take different paths to reach the destination.

### Key Characteristics

No Pre-established Path: Each packet is routed independently without a pre-established path.

Dynamic Routing: Packets may take different routes through the network based on current network conditions.

Potential Out-of-Order Delivery: Packets can arrive out of order and may need reassembly at the destination.

No Connection Setup: No initial setup phase is required.

### Process

Data Segmentation: Data is divided into packets, each containing the full destination address.

Independent Routing: Each packet is routed independently through the network, using the most efficient path at the time.

Reassembly: Packets are reassembled into the original data at the destination, which may involve reordering.

### Advantages

* Flexible and efficient use of network resources.
* More robust to changes and failures in the network as packets can be rerouted dynamically.

### Disadvantages

* Potential for out-of-order delivery and packet loss.
* More complex reassembly and error handling at the destination.

### Example

The Internet Protocol (IP) is based on datagram switching, where each IP packet is routed independently.

# Transmission Delays

In networking, the performance and efficiency of data communication are influenced by several types of delays. The primary delays include propagation delay, transmission time, and processing time. Here are the detailed differences among these:

## Propagation Delay

Propagation delay is the time it takes for a signal to travel from the sender to the receiver across the transmission medium.

### Key Factors

Distance: The length of the path between the sender and receiver.

Speed of Signal: The speed at which the signal travels through the medium, typically close to the speed of light in a vacuum, but slower in other media like copper cables or optical fibers.

### Formula

Propagation Delay = Distance / Propagation Speed

### Characteristics

Constant for a Given Medium: The propagation delay is constant for a given transmission medium and distance.

Independent of Data Size: It depends only on the physical distance and the speed of the medium, not on the size of the data being transmitted.

### Example

If the distance between two nodes is 1000 km and the signal travels at 2×108 meters per second (typical speed in fiber optics), the propagation delay would be:

Propagation Delay = 1000 ×103 meters / 2×108 meters/second = 0.005 seconds

## Transmission Time

Transmission time is the time required to push all the bits of a packet onto the wire.

### Key Factors

Size of the Data: The number of bits in the packet.

Bandwidth: The data transmission rate of the medium (bits per second).

### Formula

Transmission Time = Data Size (bits)/Bandwidth (bits per second)

### Characteristics

Depends on Data Size and Bandwidth: The larger the data size or the lower the bandwidth, the longer the transmission time.

Does Not Include Propagation Delay: It's purely the time to place the data onto the transmission medium.

### Example

If the data size is 1 megabit (1,000,000 bits) and the bandwidth is 100 megabits per second (100,000,000 bits/second), the transmission time would be:

Transmission Time=1,000,000 bits/100,000,000 bits/second=0.01 seconds

## ​Processing Time

Processing time is the time taken by the network devices (like routers and switches) to process the packet headers, check for errors, and determine the packet's next hop.

### Key Factors

Device Processing Speed: The speed at which the network device can process data.

Complexity of the Processing Tasks: Includes error checking, routing decisions, etc.

Load on the Device: The current load and queue lengths on the network device.

### Characteristics

Varies by Device and Task: Different devices and tasks will have different processing times.

Can Include Queuing Delay: If the device is heavily loaded, packets may spend additional time in queues, increasing the total processing time.

### Example

If a router processes a packet in 100 microseconds (0.0001 seconds) under normal load, but under heavy load, the queuing delay might add another 50 microseconds, the total processing time would be:

Processing Time=0.0001 seconds+0.00005 seconds=0.00015 seconds

# Assignment

*Explain the Propagation Delay, Transmission Time and Processing Time for the Packet Switching and Circuit Switching Approaches in your own words*

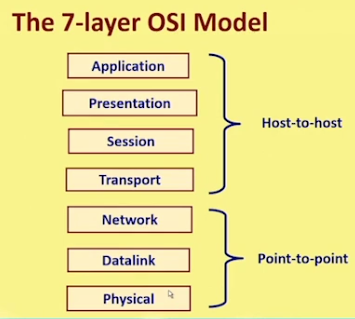
**LECTURE No. 3**

**Layered Network Architecture**

# Open Systems Interconnection (OSI) Model

The Open Systems Interconnection (OSI) model is a conceptual framework used to understand and implement network communications between computing systems. It divides the communication process into seven distinct layers, each with specific functions. Here’s a practical guide to the OSI model for beginners:

# The OSI Model Layers



**Figure 3:** Layers of OSI Model

## 1. Physical Layer

The physical layer is responsible for the transmission and reception of raw bitstreams over a physical medium. It deals with hardware components like cables, switches, and network interface cards.

### Examples

Cables: Ethernet cables (Cat5, Cat6)

Hardware: Hubs, switches

Standards: USB, IEEE 802.3 (Ethernet)

### Practical Note

This layer is all about the physical connection between devices. When setting up a network, ensuring that all cables and hardware components are properly connected and functional is crucial.

## 2. Data Link Layer

The data link layer is responsible for node-to-node data transfer and error detection/correction. It ensures that data transferred over the physical layer is error-free and reliable.

### Examples

Devices: Network switches, bridges

Protocols: Ethernet, Wi-Fi (IEEE 802.11)

Addressing: MAC (Media Access Control) addresses

### Practical Note

In a home network, the data link layer manages communication within the local network. Ensuring that devices like routers and switches are configured correctly helps in maintaining a stable network.

## 3. Network Layer

The network layer is responsible for routing data packets across networks and managing the logical addressing of devices. It determines the best path for data to travel from the source to the destination.

### Examples

Devices: Routers

Protocols: IP (Internet Protocol), ICMP (Internet Control Message Protocol)

Addressing: IP addresses

### Practical Note

When setting up internet connections, configuring IP addresses and routers correctly is essential for devices to communicate across different networks (e.g., between your home network and the internet).

## 4. Transport Layer

The transport layer ensures end-to-end communication and data transfer integrity. It manages error detection and correction and controls data flow to prevent congestion.

### Examples

Protocols: TCP (Transmission Control Protocol), UDP (User Datagram Protocol)

Functions: Segmentation, flow control, error correction

### Practical Note

When downloading a file or streaming a video, the transport layer ensures that data packets arrive correctly and in the right order. TCP is used for reliable communication (e.g., web browsing), while UDP is used for faster, less reliable communication (e.g., online gaming).

## 5. Session Layer

The session layer manages sessions or connections between applications. It establishes, maintains, and terminates communication sessions.

### Examples

Protocols: NetBIOS, RPC (Remote Procedure Call)

Functions: Session establishment, management, and termination

### ***Practical Note***

When logging into a remote server, the session layer helps establish a session for communication. Ensuring proper session management helps maintain stable connections.

## 6. Presentation Layer

The presentation layer translates data between the application layer and the network. It formats and encrypts data to ensure that it is readable by the receiving system.

### Examples:

Functions: Data translation, encryption, compression

Formats: JPEG, MPEG, ASCII, SSL/TLS (encryption)

### Practical Note

When accessing a secure website (https://), the presentation layer handles data encryption and decryption, ensuring that sensitive information like passwords is protected.

## 7. Application Layer

The application layer provides network services directly to end-users. It includes protocols that enable users to interact with the network, such as email, file transfer, and web browsing.

### Examples:

Protocols: HTTP (HyperText Transfer Protocol), FTP (File Transfer Protocol), SMTP (Simple Mail Transfer Protocol)

Applications: Web browsers, email clients, file transfer applications

### Practical Note

The application layer is where users interact with the network. Using applications like web browsers (Chrome, Firefox) and email clients (Outlook, Gmail) involves application layer protocols to access network services.

# Practical Overview

Layer 1 (Physical): Check your cables and hardware connections.

Layer 2 (Data Link): Ensure your switches and MAC addresses are properly configured.

Layer 3 (Network): Configure your IP addresses and routers correctly.

Layer 4 (Transport): Use appropriate transport protocols (TCP/UDP) based on the application.

Layer 5 (Session): Manage and maintain stable communication sessions.

Layer 6 (Presentation): Ensure data is properly formatted, encrypted, and compressed.

Layer 7 (Application): Use and configure application protocols for accessing network services.

# Transmission Control Protocol/Internet Protocol (TCP/IP) Model

The TCP/IP model is a framework used to understand and implement networking protocols. It's the basis for internet communication. To make it easier to understand, let's break it down into its four layers, using simple analogies:

## Link Layer (Network Interface Layer)

Analogy: Think of it as the postal system’s delivery vehicles (bicycles, trucks, etc.) that transport letters from one place to another within a local area.

Function: This layer is responsible for the physical transmission of data over a network medium (like cables or wireless signals). It includes hardware and device drivers. It handles the interaction with the physical hardware of the network.

## Internet Layer

Analogy: Imagine the postal system sorting facilities where letters are routed to their destination addresses.

Function: This layer is responsible for addressing, packaging, and routing functions. The key protocol here is the Internet Protocol (IP), which ensures that data packets get to the correct destination. It handles logical addressing and determines the best path for data to travel.

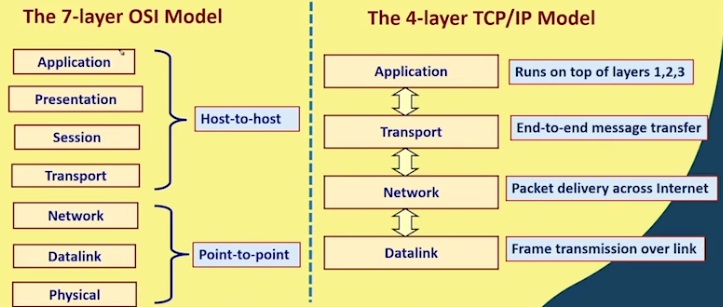
## Transport Layer

Analogy: Think of it as the postal service’s sorting and forwarding mechanisms that ensure letters are delivered in the correct order and intact.

Function: This layer provides end-to-end communication services for applications. The most common protocols here are the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP). TCP ensures reliable data transmission with error checking and recovery, while UDP allows for faster, but less reliable, communication.

## Application Layer

Analogy: Imagine the postal service’s interaction with customers, such as sending, receiving, and reading letters.

Function: This layer provides network services directly to applications. It includes protocols like HTTP (for web browsing), FTP (for file transfers), and SMTP (for email). This is the layer where user applications interact with the network.

**Figure 4:** Comparison b/w TCP/IP and OSI Model

## How They Work Together

When you send an email or browse a website, data is passed down through these layers, each adding its own information:

Application Layer: Your email program (application) creates the message.

Transport Layer: TCP divides the message into segments and ensures they are delivered accurately.

Internet Layer: IP packets are created with the correct destination address and routing information.

Link Layer: The data is sent over the physical network (e.g., Ethernet or Wi-Fi) to its destination.

On the receiving end, the process is reversed:

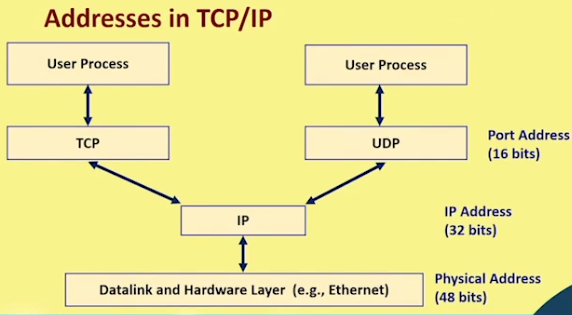
Link Layer: Receives the data and passes it up to the Internet Layer.

Internet Layer: Routes the packets to the correct device.

Transport Layer: Reassembles the segments into the original message.

Application Layer: The email program displays the message to the recipient.

## Why It Matters

Understanding the TCP/IP model helps explain how data travels across the internet and how different networking components interact. It’s the backbone of modern communication, making it essential for anyone interested in networking or internet technologies.

**Figure 5:** Addressing Bits on Each Layer of TCP/IP Model

**LECTURE No. 4**

**IP Datagram and Header Fields**

Imagine you want to send a letter to a friend who lives in a different city. You put the letter in an envelope, write your friend's address on it, and then drop it in a mailbox. The postal service takes care of delivering it to your friend's address.

Now, think of the Internet as a huge postal system that delivers digital information. Instead of letters, we're sending data (like emails, web pages, videos, etc.) from one computer to another. This data needs to be packed up and addressed correctly so it can travel across the Internet.

Here's where the IP datagram comes in:

Data Package

When you want to send some data over the Internet, it gets divided into smaller chunks. Each chunk is like a letter you want to send.

# Envelope (Datagram)

Each chunk of data is put into its own digital "envelope" called an IP datagram. Just like a real envelope, the IP datagram has important information written on it to make sure it gets to the right place.

# Address (IP Address)

The IP datagram has a source address (where it's coming from) and a destination address (where it's going). These addresses are called IP addresses and work like street addresses in the postal system.

# Labelling (Headers)

Besides the addresses, the datagram also has some additional information written on it, like a label that helps it find its way. This label includes things like the type of data being sent and how to put all the chunks back together once they arrive.

# Delivery (Routing)

Once the IP datagram is ready, it gets sent into the Internet. Just like the postal system, the Internet has many routers (like post offices) that read the addresses on the datagram and help direct it towards its destination.

# Reassembly

When all the datagrams arrive at their destination (your friend's computer), they are reassembled back into the original data (like reassembling a torn-up letter) so your friend can read it.

## So, in summary:

* An IP datagram is like a digital envelope that carries a chunk of data across the Internet.
* It has a source address and a destination address, which are like the return address and the recipient's address on a real envelope.
* Routers act like post offices, helping the datagram find its way to the destination.
* Once all the datagrams arrive, they are put back together to form the original message.

This process happens very quickly and invisibly every time you use the Internet, making it possible for us to browse websites, send emails, and watch videos seamlessly.

# IP Datagram’s Headers

The IP datagram's headers, like those sticky notes on our internet envelope analogy, carry a specific set of information to guide its journey across the web. Here's a breakdown of the main mandatory fields:

1. Version: This is like a sticker saying "IP Version 4" (most common) or "IP Version 6" (newer version). It tells routers what kind of IP language the datagram speaks.
2. Header Length (IHL): Imagine a note saying "20 bytes" or "40 bytes." This tells routers how long the header section is, including all the sticky notes!
3. Type of Service (TOS): This is like a note saying "Priority," "Standard," or "Low Cost." It lets routers know how important the datagram is and if it needs special treatment, like faster delivery for a video call.
4. Total Length: This sticky note says "500 bytes" or "1000 bytes," indicating the total size of the entire datagram, including both the header and the information it carries.
5. Identification (ID): Think of this as a tracking number for the datagram. It helps routers identify and manage multiple datagrams sent from the same source.
6. Flags: These are like a few checkboxes on the envelope. They tell routers if the datagram can be broken down into smaller pieces for faster travel (Don't Fragment flag) or if more fragments are coming (More Fragments flag).
7. Fragment Offset: If the datagram is broken down, this sticky note tells the router where this specific piece fits in the entire puzzle, like "Piece 2 of 3."
8. Time to Live (TTL): Imagine a note saying "Hops: 5." This sets a limit on how many routers the datagram can travel through before it gets discarded. It prevents datagrams from circling around the internet forever.
9. Protocol: This note tells the router what type of information the datagram is carrying inside. Is it a website (TCP) or a streaming video (UDP)?
10. Header Checksum: This is like a complex code on the envelope. The sender calculates this code based on the header information, and the receiver checks it again to ensure the header data hasn't been corrupted during travel.
11. Source Address & Destination Address: These are the big ones, like the "From" and "To" addresses on a regular envelope. They tell the internet where the datagram came from and where it needs to be delivered.

# Optional Fields:

There's also an optional section with space for additional sticky notes if needed. These might include things like security options or timestamps.

# IP Addressing and Routing

## Fragmentation

IP packet fragmentation is a process that allows large packets to be broken down into smaller pieces so they can be transmitted over networks that have size limitations on the packets they can handle. Here's a simple explanation:

## Why Fragmentation is Needed

When data is sent over a network, it is broken down into smaller units called packets. Each packet has a size limit, which is determined by the network's Maximum Transmission Unit (MTU). If a packet is larger than the MTU, it cannot be transmitted as is and needs to be fragmented.

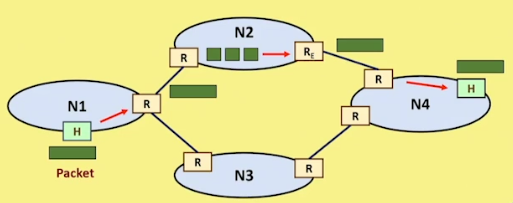
Based on the Fragmentation Process, it is divided into two parts

### Transparent Fragmentation

In transparent fragmentation, the fragmentation process is hidden from subsequent networks. This means that once a packet is fragmented by a router, the fragments are reassembled at the exit point of the same network before being forwarded to the next network.

#### Key characteristics:

* The entire fragmentation and reassembly process occurs within a single network.
* Subsequent networks are unaware of the fragmentation that took place.
* Reduces the complexity of handling fragments in intermediate networks.

Example: ATM networks often employ transparent fragmentation.

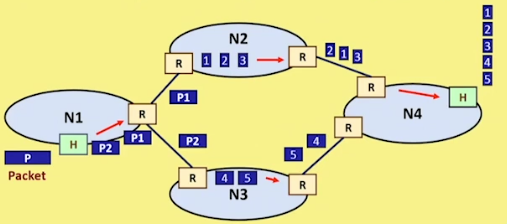
**Figure 6:** Transparent Fragmentation

### Non-Transparent Fragmentation

In non-transparent fragmentation, fragments are forwarded to subsequent networks without being reassembled. Each network along the path may potentially fragment the packet again if its MTU is smaller. The final reassembly occurs at the destination host.

#### Key characteristics:

* Fragments travel through multiple networks without being reassembled until they reach the destination.
* Increased complexity in handling fragments as each network needs to manage them.
* Potential for multiple fragmentations, increasing overhead.

Example: The internet primarily uses non-transparent fragmentation.

**Figure 7:** Non-Transparent Fragmentation