Ports, Protocols, and the OSI Model

Cybersecurity

Networking 101, Day 2

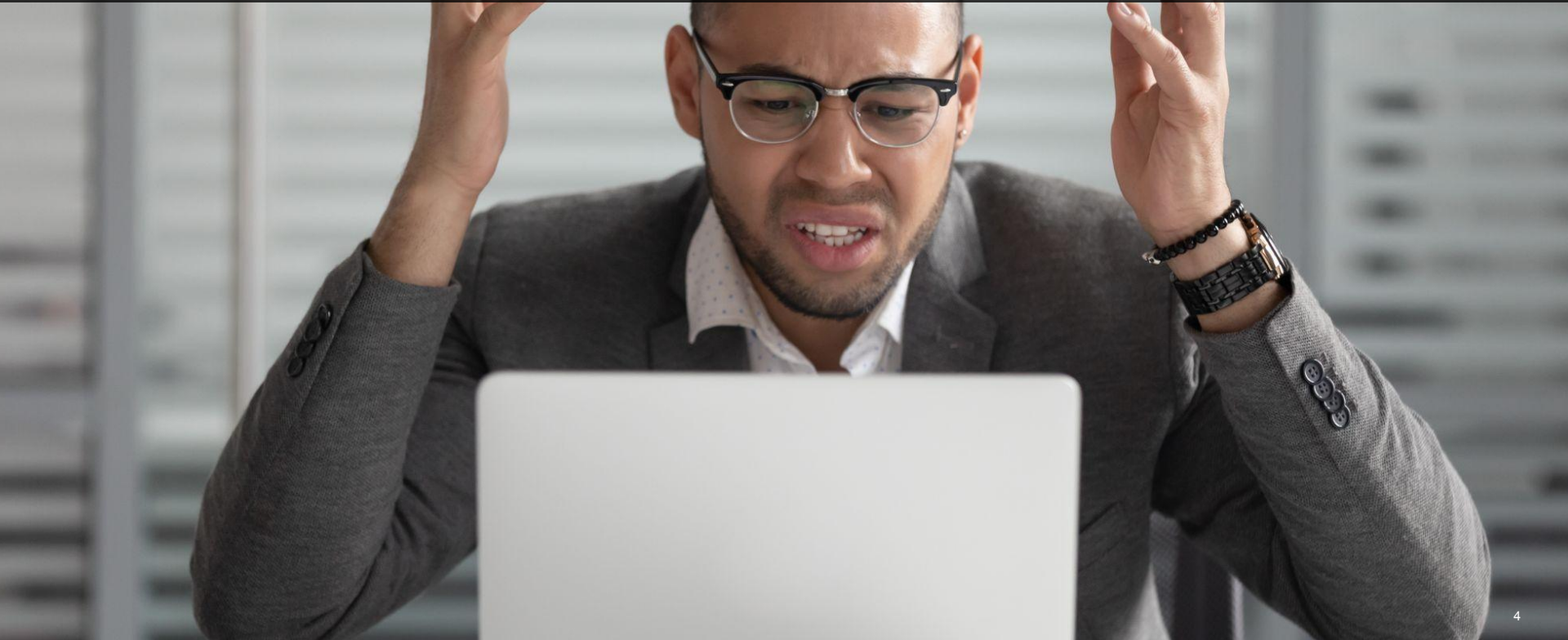
Class Objectives

By the end of class, students will be able to:

- Interpret data in network packets by analyzing their headers, payloads, and trailers.
- Explain the role of ports in specifying a network packet's destination.
- Associate common protocols with their assigned ports.
- Explain how encapsulation and decapsulation allow different protocols to interact with one another.
- Use the layers of the OSI model to identify sources of problems on a network.
- Capture and analyze live network traffic using Wireshark.

Protocols
“*Roger, Over, and Out*”

With so much going on behind-the-scenes in network communication, it’s important to have systems in place to avoid the many potential errors.



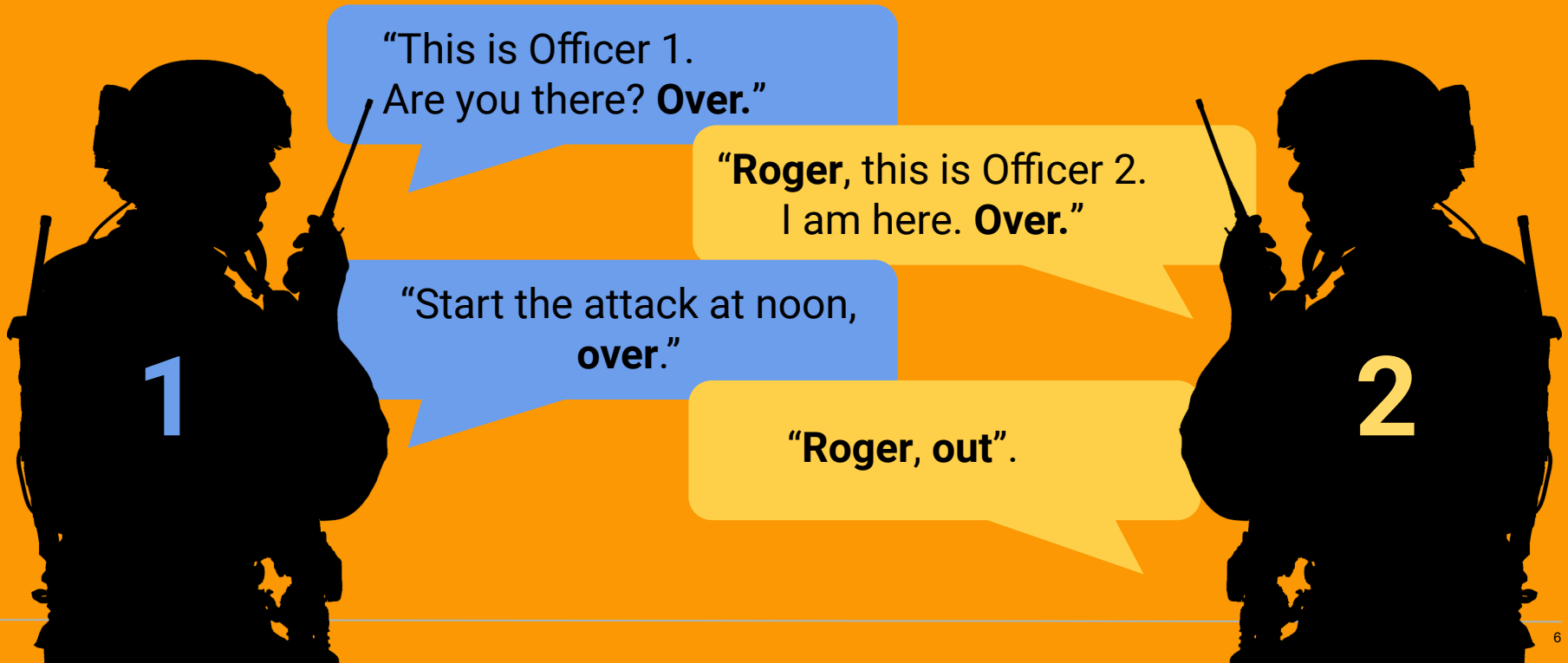
Real World Protocols

When military officers exchange messages over a radio, they can encounter the following issues:

- Communications don’t come in clearly.
- Multiple communications are sent at the same time.
- Communications are cut off.

Real World Protocols

Military personnel use **communication protocols** to ensure messages are *transmitted*, *received*, and *understood* clearly.



Real World Protocols

This mode of communication ensures that there is no ambiguity when sending and receiving messages. Clear meanings for certain keywords and strict rules for using them drastically improves the efficacy of communications.

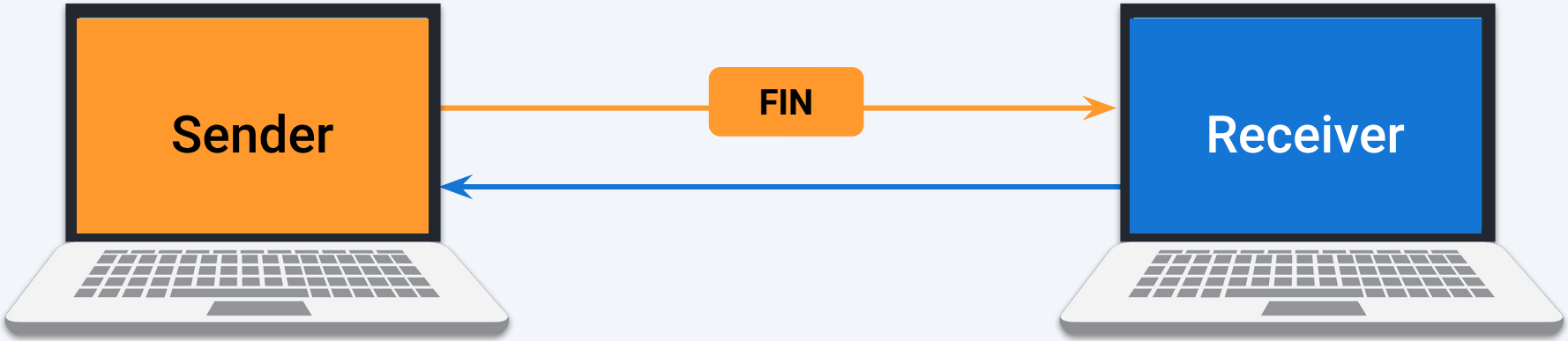
“Over”	“Roger”	“Out”
Signals the end of a specific line of communication.	Signals that the message was received completely.	Signals that the message was received, the exchange is complete, and the order will be followed.

These strict rules are known as a **protocol**. They impose structure by specifying the precise meaning of keywords, and where in a message they must appear.

Protocols and Networks

Networks use protocols to ensure messages are fully sent and understood.

Similar to the military's use of "over," a network uses the TCP message **`FIN`** to indicate the end of the transmission.



Networking Protocols

Some common protocols you may be familiar with:

HTTP

(Hypertext Transfer Protocol)

Used to communicate web traffic.

FTP

(File Transfer Protocol)

Used to transfer files.

Networking Protocols

Some other important protocols:

PAP	*(Password Authentication Protocol)*	Used for authenticating a user.
SMB	*(Server Message Block)*	A Windows-based protocol used for sharing files.
NetBIOS	*(Network Basic Input/Output System)*	Allows computers to communicate on a local network.

There are often multiple protocols available for the same type of task. In these cases, you will choose which protocol to use based on context.

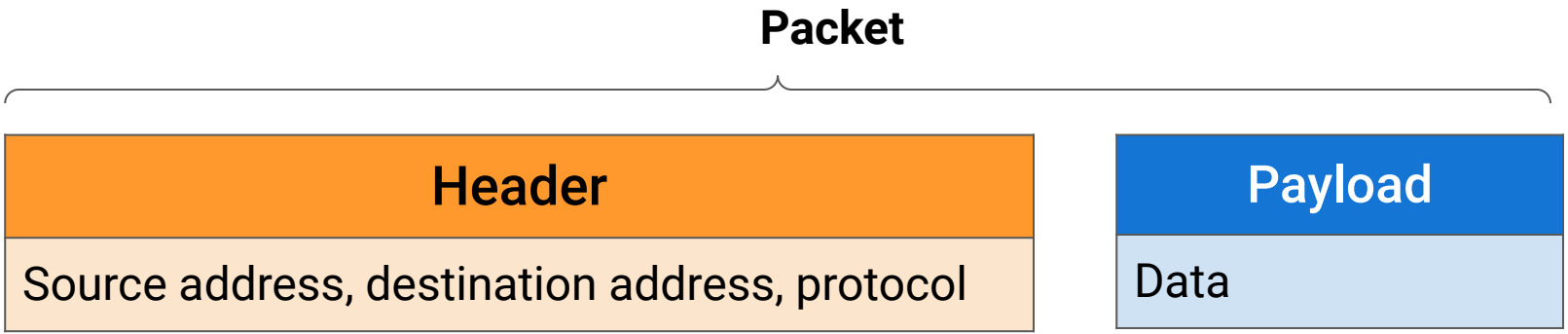
Network Packet Structure

Protocol rules impose structure on communications. A message adhering to protocol rules is called a **network packet**.

As we know from the previous class, the client and server communicate by exchanging binary data.

IPv4 Packets

The binary data is grouped together into separate pieces, known as **packets**, and transmitted across the network.



Packet Size (up to 65,535 bytes) = Header (20-60 bytes) + Payload

IPv4 Packet Structure

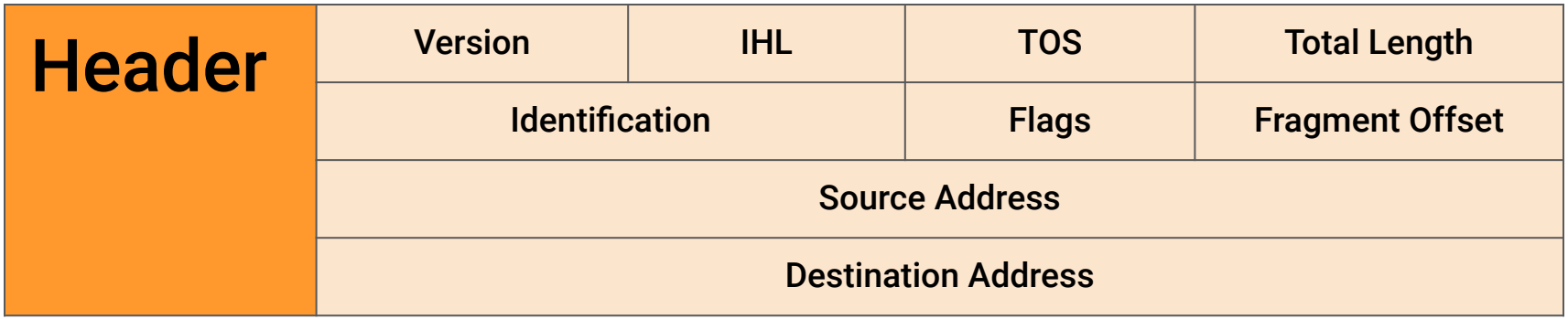
The basic structure of an IPv4 packet has the following components.

A packet contains the data that it is sending and information pertaining to how the packet should be handled.

<table>
<tr><td rowspan="6">Header</td><td>Version</td><td>IHL</td><td>TOS</td><td>Total Length</td></tr>
<tr><td colspan="2">Identification</td><td>Flags</td><td>Fragment Offset</td></tr>
<tr><td colspan="2">TTL</td><td>Protocol</td><td>Header Checksum</td></tr>
<tr><td colspan="4">Source Address</td></tr>
<tr><td colspan="4">Destination Address</td></tr>
<tr><td colspan="4">Options</td></tr>
<tr><td>Payload</td><td colspan="4">Data</td></tr>
</table>

IPv4 Packet Header

The header of an IPv4 packet contains the information to help the receiver understand how to handle the packet. For example, the sender and the receiver of the packet are contained within the header.



The specific arrangement of packets allows the receiving party to properly interpret the contents and direction of the communication.

IPv4 vs. IPv6 Packet Header

An IPv6 Packet is similar, but there are differences. For instance, an IPv6 Packet Header has the following components.

IPv4

Version	IHL	TOS	Total Length
Identification		Flags	Fragment Offset
Source Address			
Destination Address			

IPv6

Version	Traffic Class	Flow Label	
Payload Length		Next Header	Hop Limit
Source Address			
Destination Address			

Packets

The header may contain **fields** such as:

Protocol	Specifies the name of the protocol in use.
Version	Specifies which version of the protocol is in use.
Destination address	Specifies where the packet is going.
Length of packet	Tells the receiver how much data is in the packet payload.

- Each of these parts and the fields within them will appear in an agreed-upon order and size so that the receiver knows exactly where to find specific information.
- This order is dictated by the rules of the protocol in use.

Packets

These parts and the fields within them will appear in an agreed-upon order and size so that the receiver knows exactly where to find specific information. This order is dictated by the rules of the protocol in use.

All protocols contain a header and a payload.

Not all protocols include a trailer

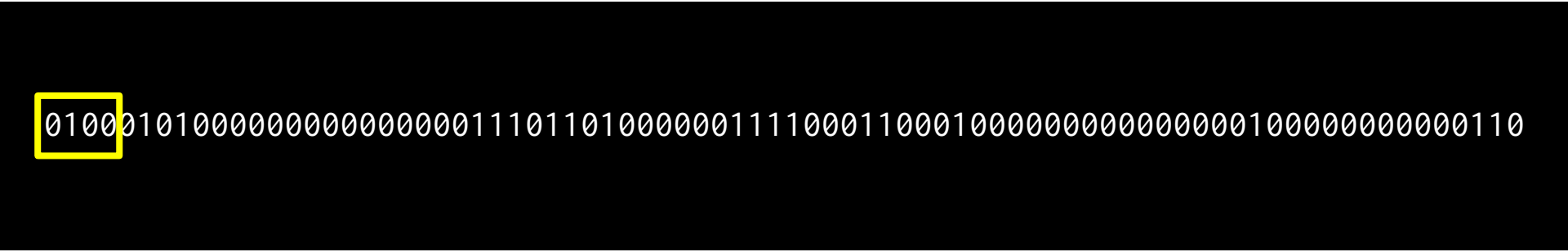
Different protocols often contain different header and payload fields.

Packets Example

The **version** field is indicated in the header.

As the first field, it starts at the first bit and ends at the fourth bit.

The receiver will always find this information in this exact location.



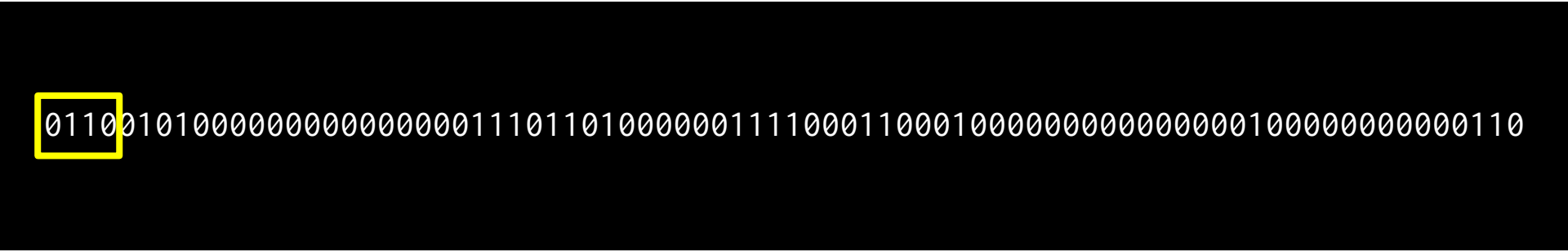
In binary, **0100** translates to **4**.
Therefore, this field indicates that this header uses **Version 4**.

Packets Example

The **version** field is indicated in the header.

As the first field, it starts at the first bit and ends at the fourth bit.

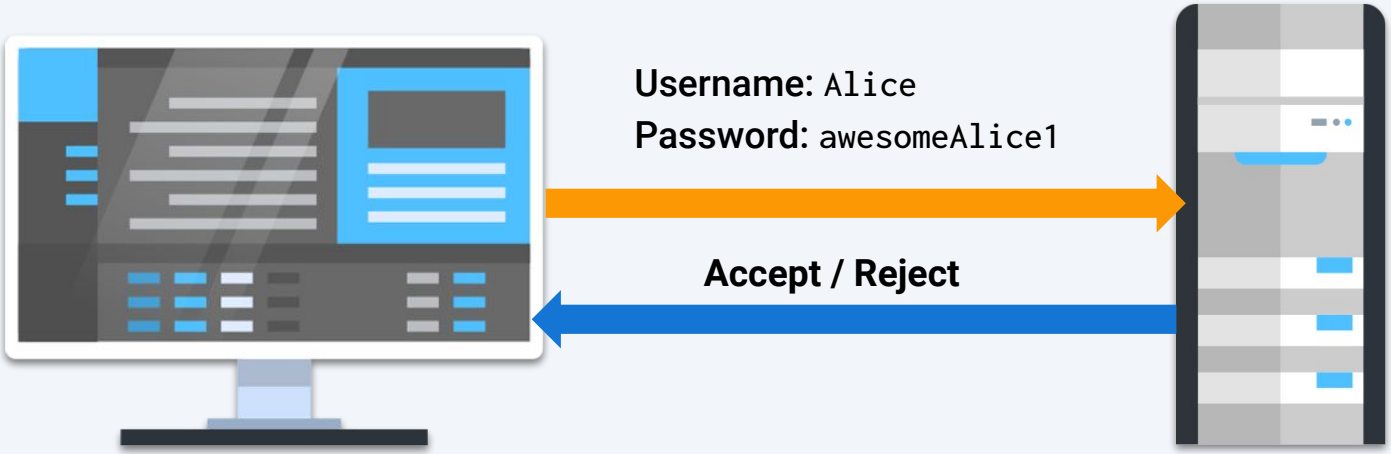
The receiver will always find this information in this exact location.



In binary, **0110** translates to **6**.
Therefore, this field indicates that this header uses **Version 6**.

Protocol Example: Authentication

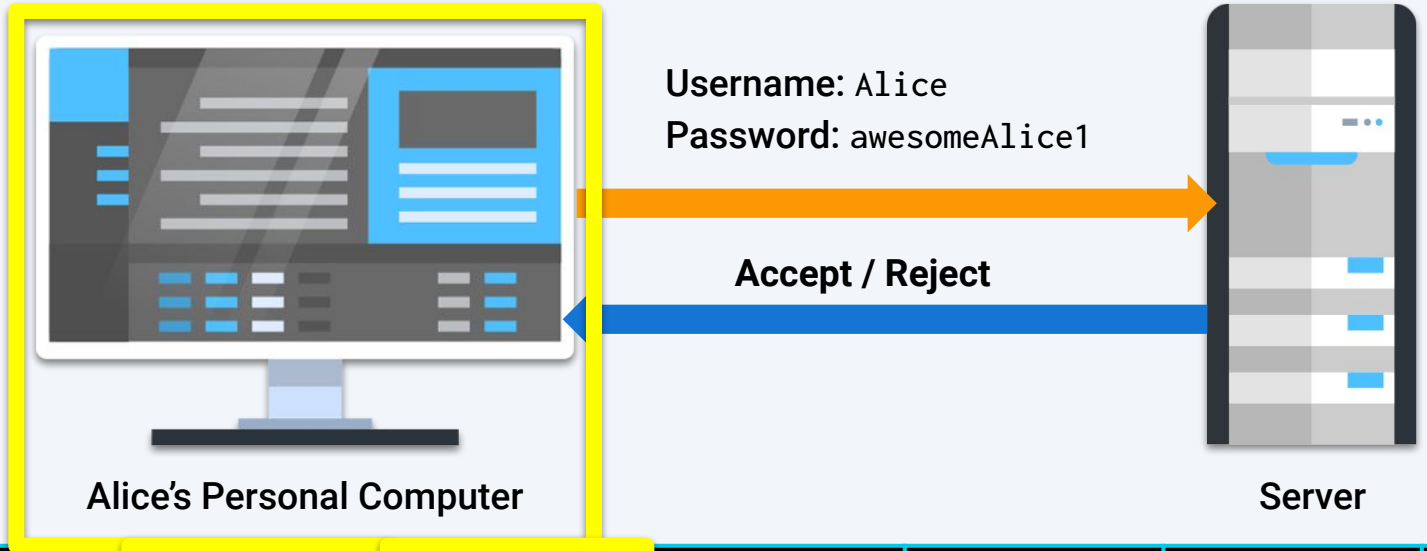
PAP (*Password Authentication Protocol*)



PAP Two-Way Handshake

Protocol Example: Authentication

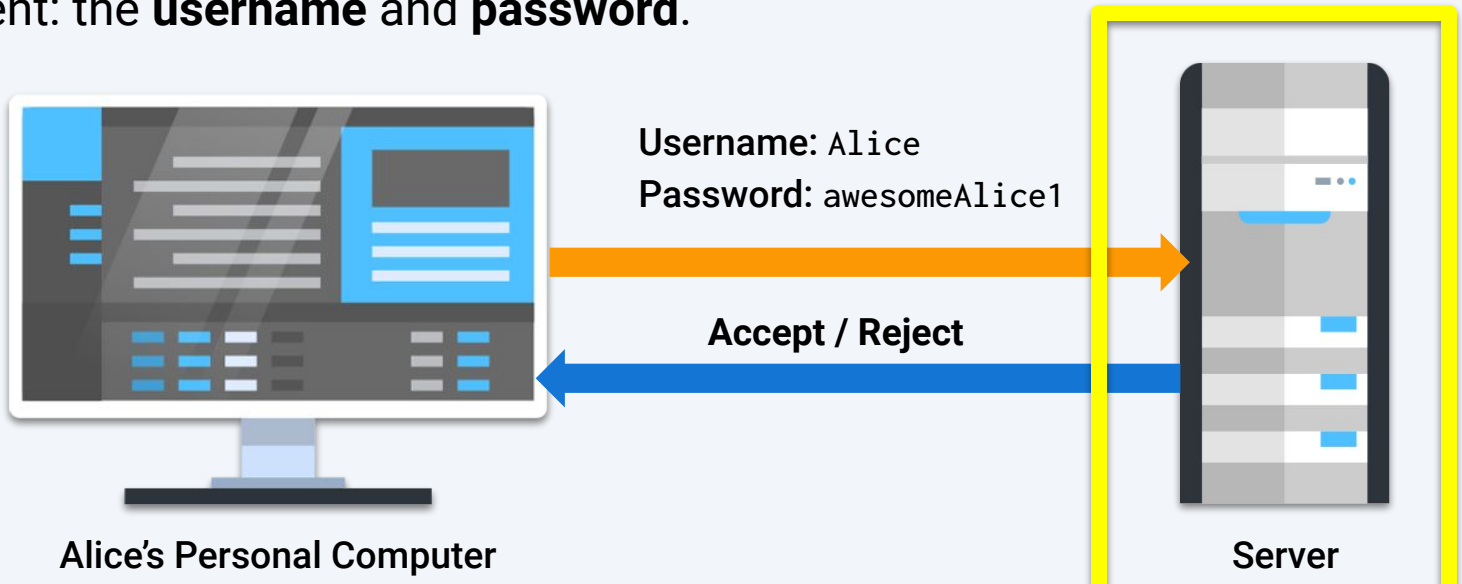
Client request contains bits in a specific **order** and **length**, per the standard and rules of the protocol.



1 Byte	1	2 Bytes	1 Byte	Variable	1 Byte	Variable
Code: 1	ID	Length	User name length	Username	Password Length	Password

Protocol Example: Authentication

The server receiving the request will know where to look in the bitstream for content: the **username** and **password**.

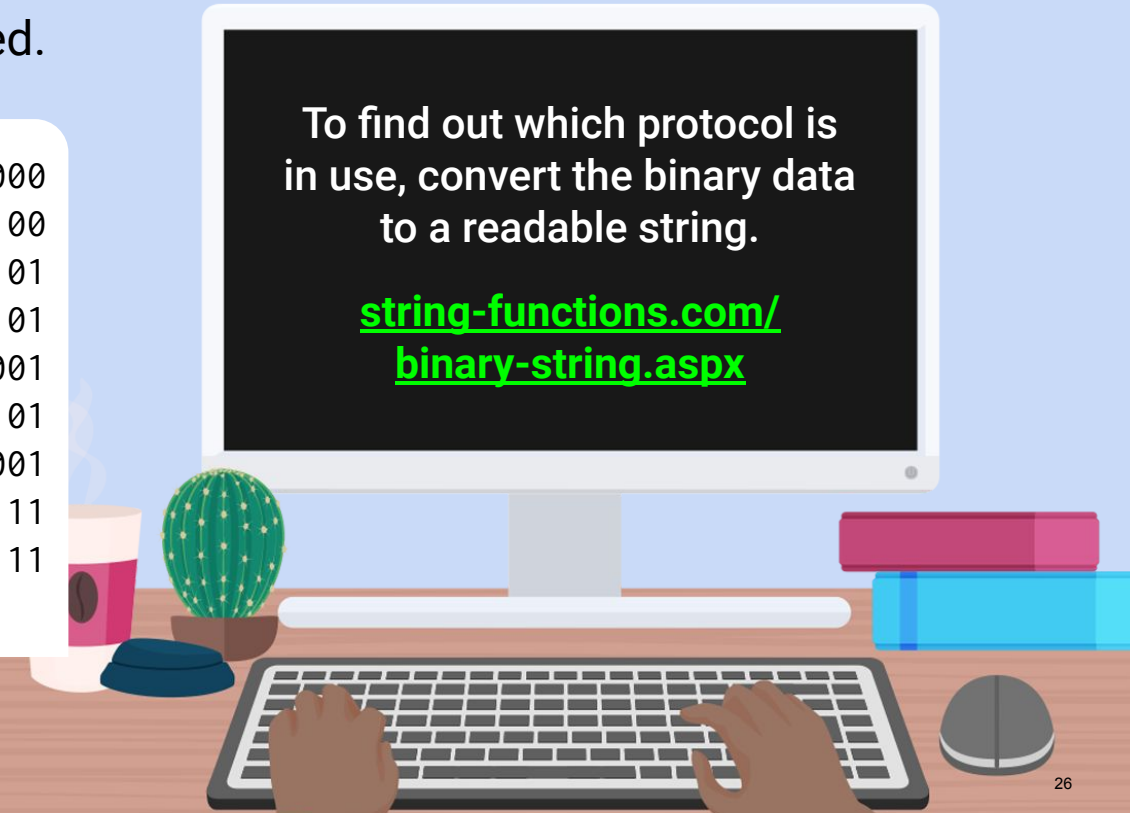


1 Byte	1	2 Bytes	1 Byte	Variable	1 Byte	Variable
Code: 1	ID	Length	User name length	Username	Password Length	Password

Interpreting Protocols from Raw Binary Data

Security professionals rely on web tools to convert binary data and determine the protocol being used.

```
01110000011000010111000000100000
01110011011001010110111001110100
00101101011101010111001101100101
01110010011011100110000101101101
01100101001000000101000001000001
01010000010101010101001101000101
01010010001000000111000001100001
01110011011100110111011101101111
01110010011001000010000000110111
```



Activity: Interpreting Protocols

In this activity, you will continue to play the role of a security analyst at Acme Corp.

Your task is to convert the raw binary data into a readable format and determine which protocol is being used.

Suggested Time:

15 Minutes

Time's Up! Let's Review.

Questions?

Ports

Ports Real World Analogy

Bob wants to show a video presentation to Alice, a coworker who works across town. Bob will be presenting from Room 33, the video conference room in his office building.

- Bob told Alice to meet him at his office, but only gave her the building address, 150 Main Street.
- Because Bob didn't specify which room he'd be in, Alice will be able to find the office building, but not Bob or the video presentation.

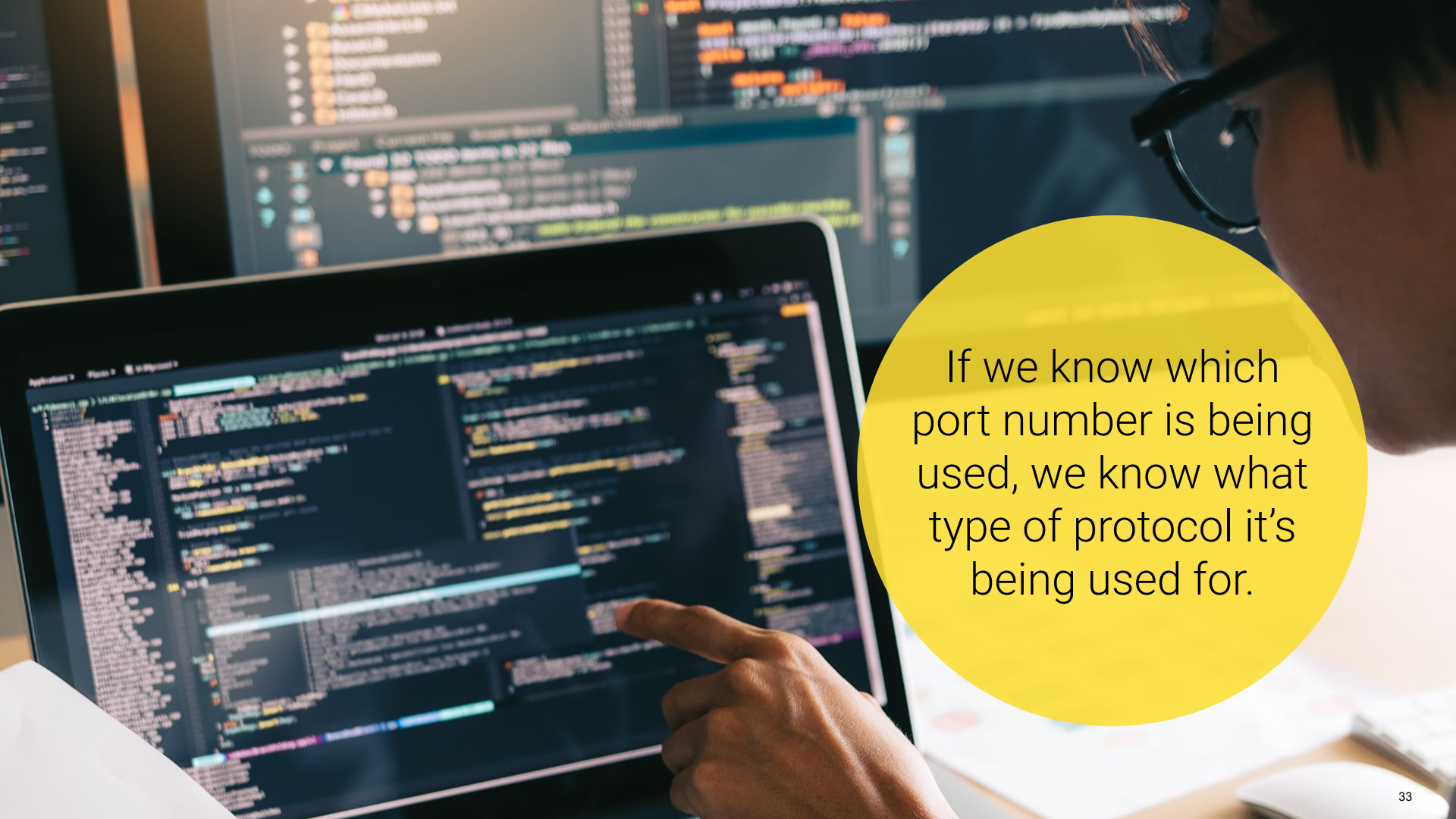
Ports Real World Analogy

In this analogy, the address of the office building is the **IP address**.

The video conference room is the **port**.

Since we know that Room 33 is the video conference room, we know that any meeting in Room 33 will involve video conferencing, even if we're not explicitly told what to expect from the meeting.

Similarly, port numbers can be associated with a specific network function and protocol.


If we know which port number is being used, we know what type of protocol it’s being used for.

Ports and Security

Ports are the access points for transmitting and receiving data.

- Ports are like doors that can be open, closed, or accessible to certain individuals.
- Since a port is considered an access point into a system, it is important that IT professionals do not allow unauthorized access to them.
- Unauthorized access can potentially lead to a breach.

The **Internet Assigned Numbers Authority (IANA)** is the entity officially responsible for assigning port numbers for a designated purpose.

Unofficial port numbers can also be used for any protocol or service, but this practice goes against the typical IANA guidelines.

https://www.iana.org/

Port Numbers

There are 65,536 virtual ports, numbered from `0` to `65535`.
These ports are divided into three ranges:

01 System ports

02 Registered ports

03 Dynamic / private ports

Virtual Ports

Computers don’t have enough physical space for every protocol, so we use software to create **virtual ports**.

Every protocol is assigned a numerical virtual port number.

The corresponding port is the destination port. It’s where other machines send data to communicate with that protocol.

For example:

A machine sending an HTTP message to a web server sends traffic to the server’s port `80`.

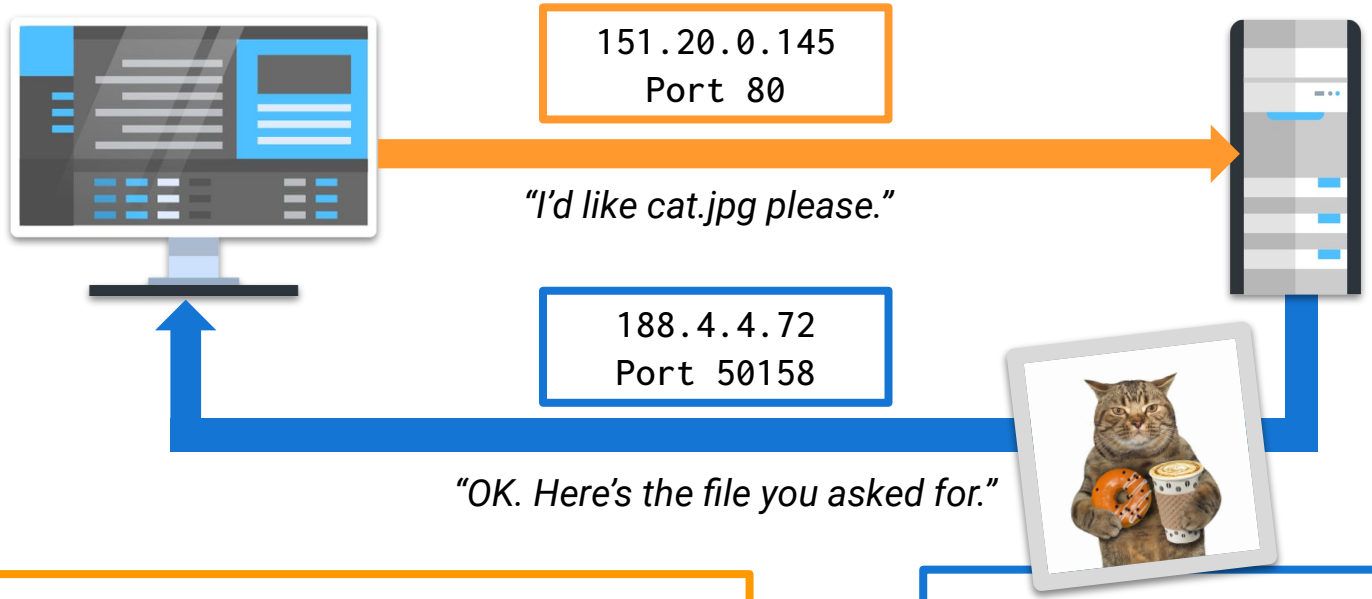
Port Numbers

Range Title	Range Number	Also Known As	General Explanation
System Ports	0–1023	Well-known ports	Restricted: Only the operating system or administrators can bind services to these ports. E.g., HTTP typically runs on port 80. Normal users can't launch services on port 80, so we can trust that machines accepting connections to port 80 are using it to send and receive HTTP traffic.
Registered Ports	1024–49151	User ports	“Normal” users launching their own services will do so using ports in this range.
Dynamic Ports	49152–65535	Dynamic ports/private ports	When a machine sends data to another machine, it must open a port to send from. This is called a source port. Source ports are randomly chosen from the dynamic range whenever a machine sends a message.

Common Ports

Port 80	HTTP	Sending web traffic.
Port 443	HTTPS	Sending encrypted web traffic.
Port 21	FTP	Sending files.
Port 22	SSH	Securely operating network services.
Port 25	SMTP	Sending emails.
Port 53	DNS	Translating domains into IP addresses.

Source / Destination Ports



The client (initiator) has a **source** port. Source ports are randomly generated from the dynamic port range (49152 - 65535).

The server (receiver), has a **destination** port, depending on the protocol. Destination ports don't change, so we can associate a port with a certain protocol.


Recap

Recap

Let’s review ports:

IP addresses are used to locate a computer on a network, such as the internet.

A port number is used to locate a specific service on that computer.

Both numbers are required to transport data.

Any service can be run on any port, but certain services have associated ports that they're expected to run on.

Activity: Ports

In this activity, you will continue to play the role of a security analyst at Acme Corp.

Your task is to determine the **source** and **destination** port for each request as well as the protocol for each destination port.

You must then research the protocol to determine what kind of activities the rogue employee may be conducting.

Suggested Time:

15 Minutes

Time's Up! Let's Review.

Questions?

Countdown timer

15:00

(with alarm)

OSI Layers


When data travels across a network, it goes through multiple steps and processes to reach its destination.

OSI Layers

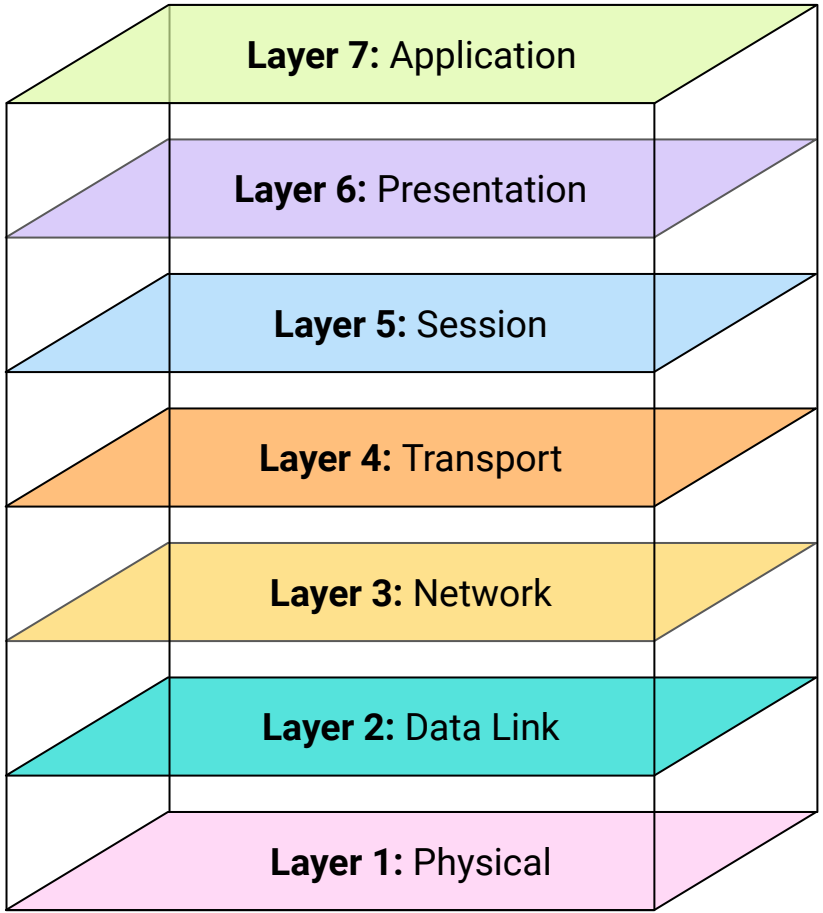
	01	02	03
Example: **The process of sending an email starts with the following steps:**	Convert the text of the user's email into a format the email application can understand.	Add the destination address and destination port to this data.	Convert this packet to a format that can be transmitted through physical wires.
Certain protocols are responsible for handling specific steps of data transmission:	The *IMAP* or *POP3* converts the text of a user's email into a format any email application can understand.	The *IP* adds destination address information to the email data.	The *TCP* adds information about destination ports to this data.

Since many of these steps are common across different scenarios, many protocols often perform the same “general step.”

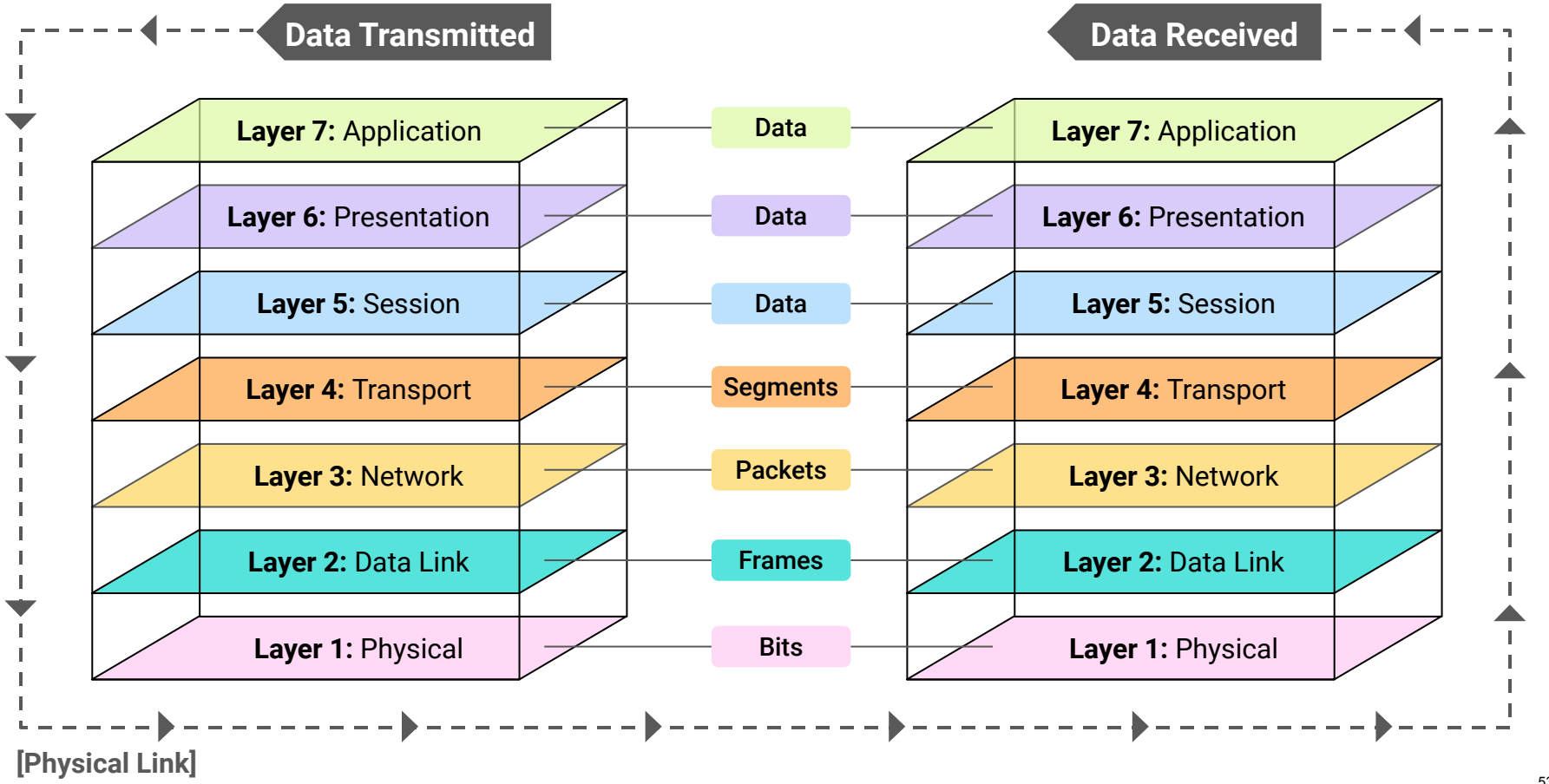
The **Open Systems Interconnection (OSI)** model is a framework that conceptualizes the steps data takes during transmission, making it easier to communicate where issues exist.

Open Systems Interconnection (OSI) Model Framework

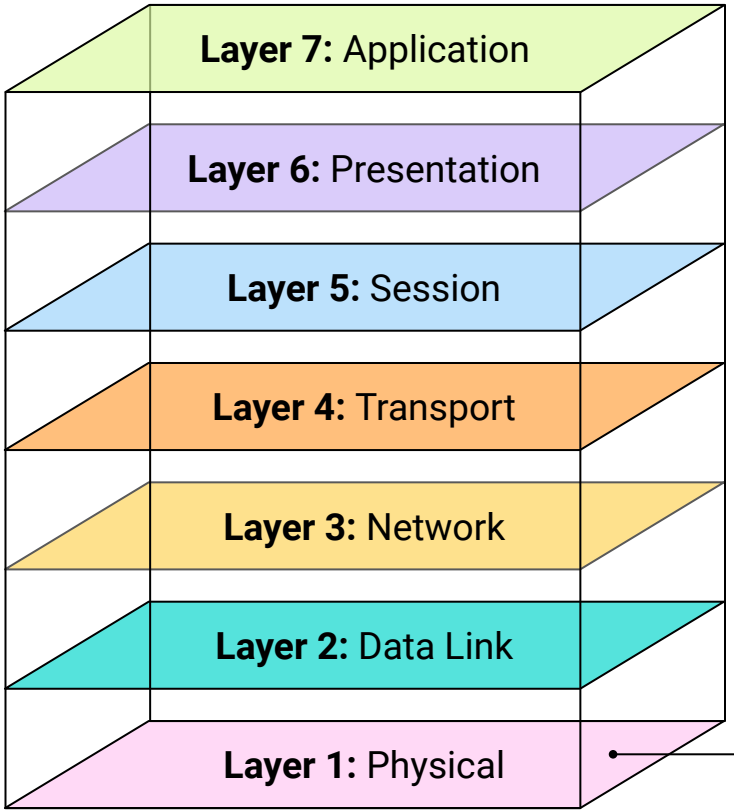
- Conceptualizes the steps data takes during transmission, making it easier to communicate where issues exist.
- Allows security analysts to better understand how communication works on a network by detailing the processes, devices, and protocols in place at each layer.



OSI Model


Data Transmitted
Data Received
Layer 7: Application
Layer 6: Presentation
Layer 5: Session
Layer 4: Transport
Layer 3: Network
Layer 2: Data Link
Layer 1: Physical
Data
Data
Data
Segments
Packets
Frames
Bits
Layer 7: Application
Layer 6: Presentation
Layer 5: Session
Layer 4: Transport
Layer 3: Network
Layer 2: Data Link
Layer 1: Physical
[Physical Link]

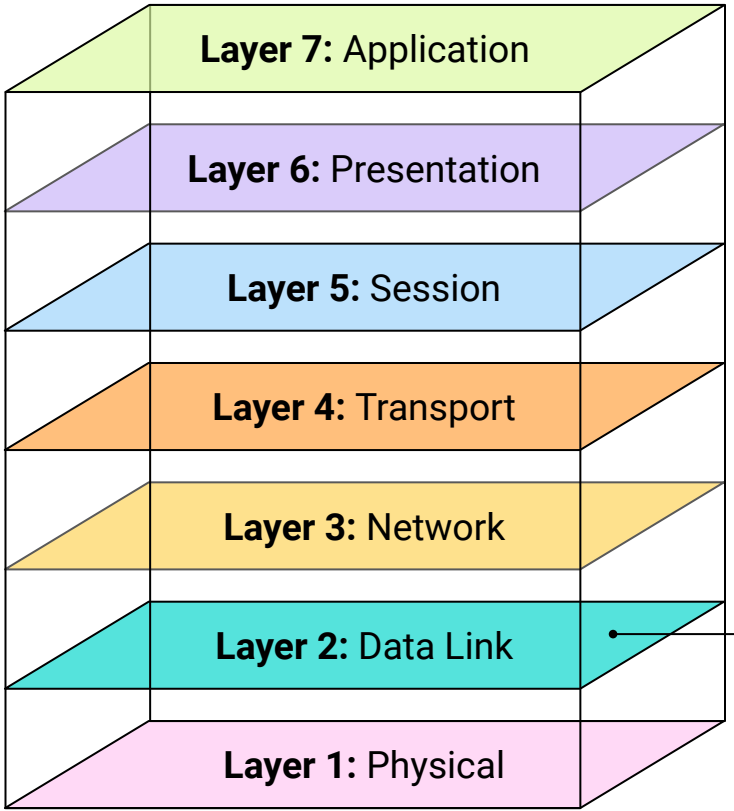
Layer 1: Physical



The **Physical layer** is responsible for transmission of binary data through a physical medium.

It handles how data is physically encoded and decoded.

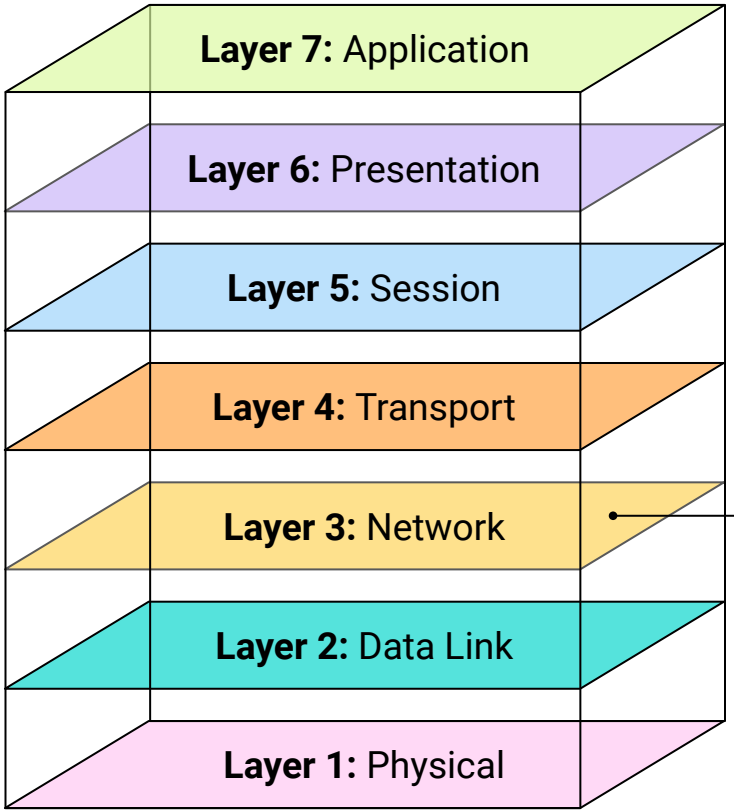
Layer 2: Data Link



The **Data Link layer** establishes links between nodes.

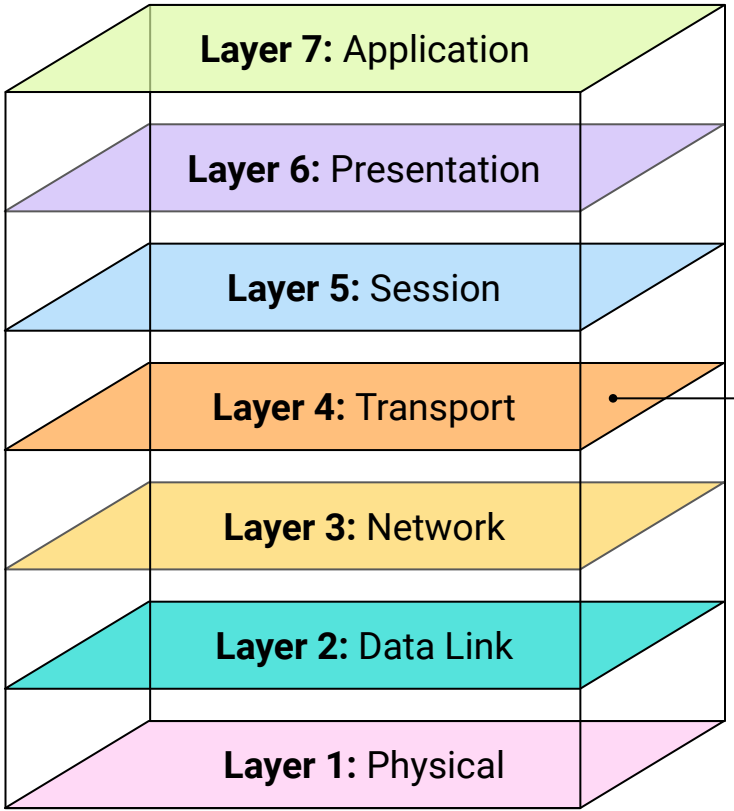
It also ensures data gets to its final destination without corruption, thus protecting data integrity.

Layer 3: Network



The **Network layer** routes data through physical networks using an IP address, deciding which physical path the data will take, and ensuring it gets to the correct destination.

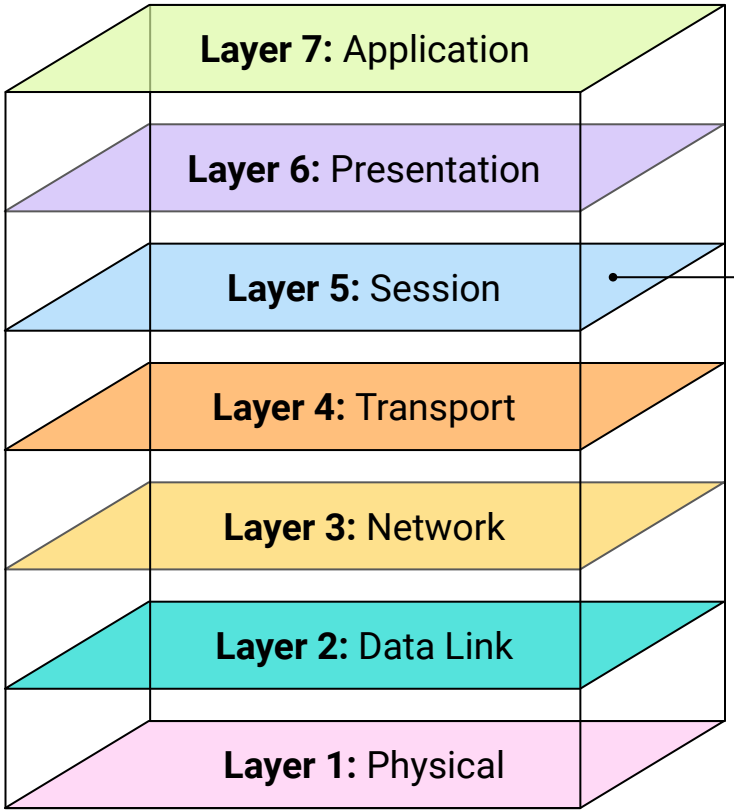
Layer 4: Transport



The **Transport layer** is responsible for actually transmitting data across the network.

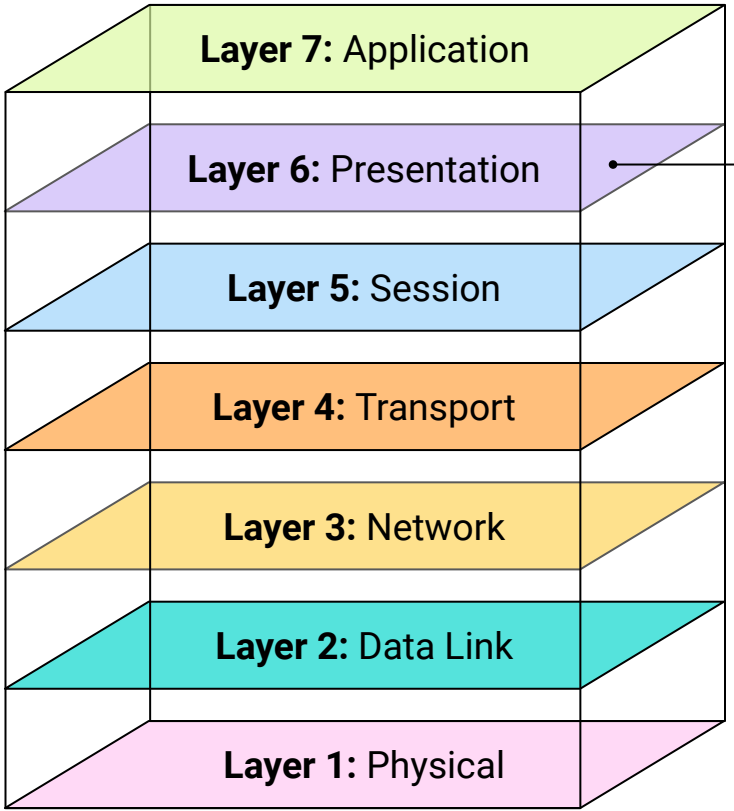
It puts data onto the network, and assigns source and destination ports.

Layer 5: Session



The **Session layer** manages connections between ports on computers and handles data flow.

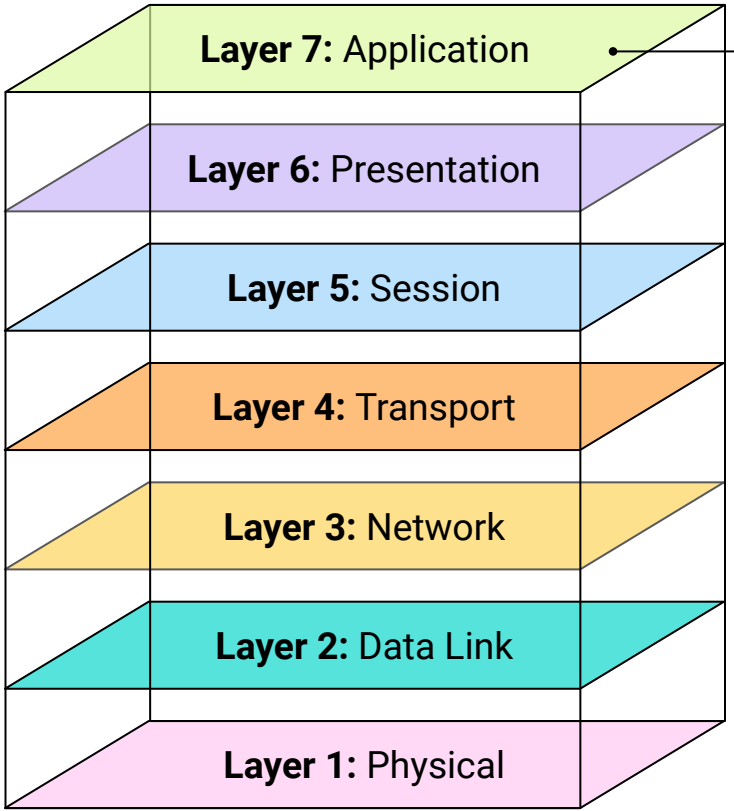
Layer 6: Presentation



The **Presentation layer** is is the translator for the network.

It formats data to be presented to the Application layer, handles data representation, decryption and encryption, character set translation, and conversion.

Layer 7: Application



The **Application layer** is responsible for representing data in a way the consuming application can understand.

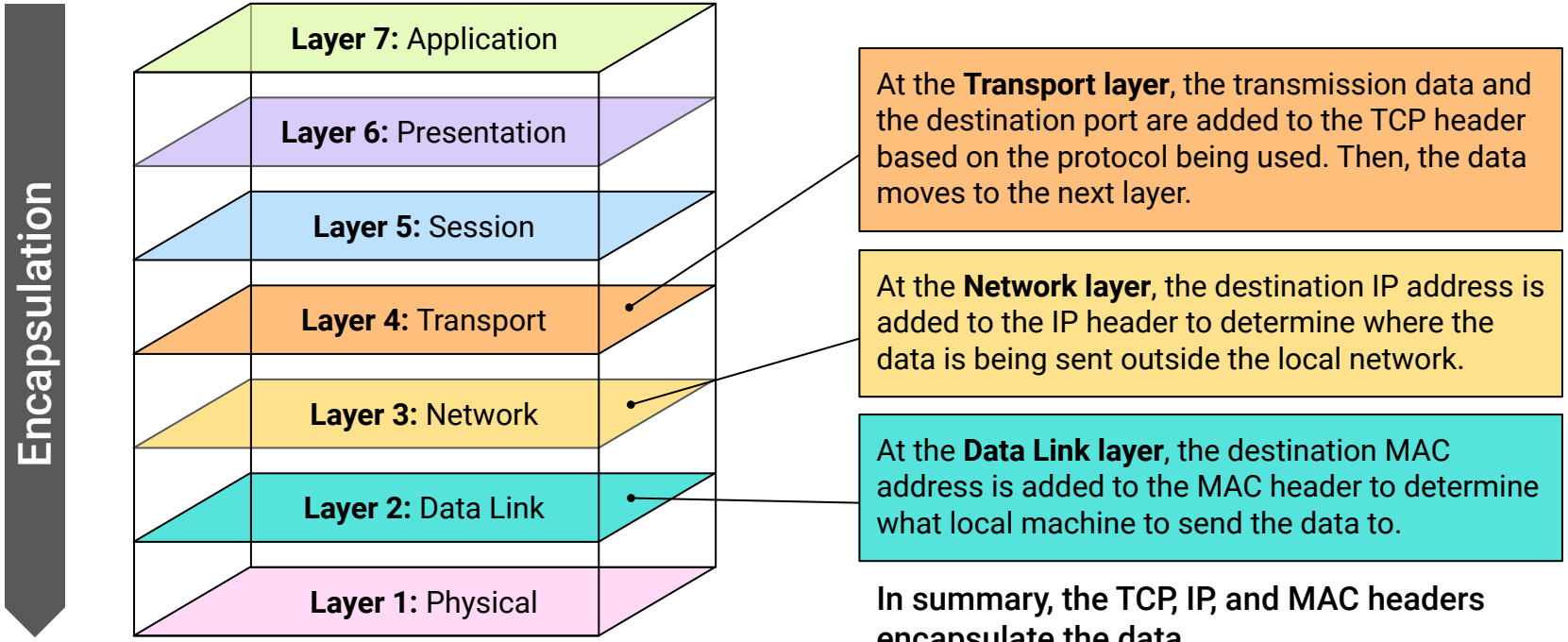
This is the layer a user interacts with, such as a web or email application.

Encapsulation and Decapsulation

Encapsulation is the process of placing headers, and sometimes trailers, around the data to direct it to its destination.

Encapsulation

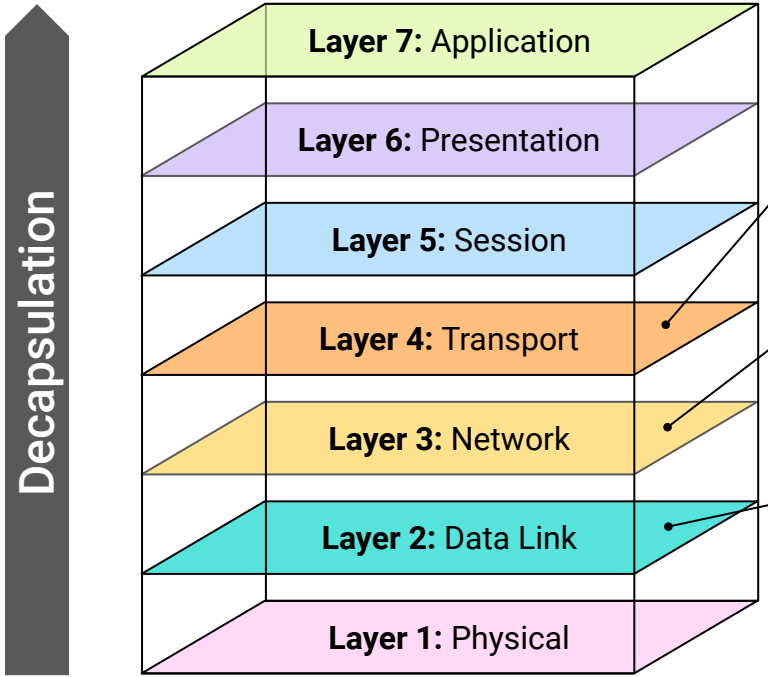
As data moves through the layers, starting from Layer 7 and ending at Layer 1, the data is **encapsulated**.



Decapsulation is the process of removing the headers, and sometimes trailers, around the data to confirm the data has reached the destination.

Decapsulation

The following examples of **decapsulation** occur as data is moving across layers in reverse order:



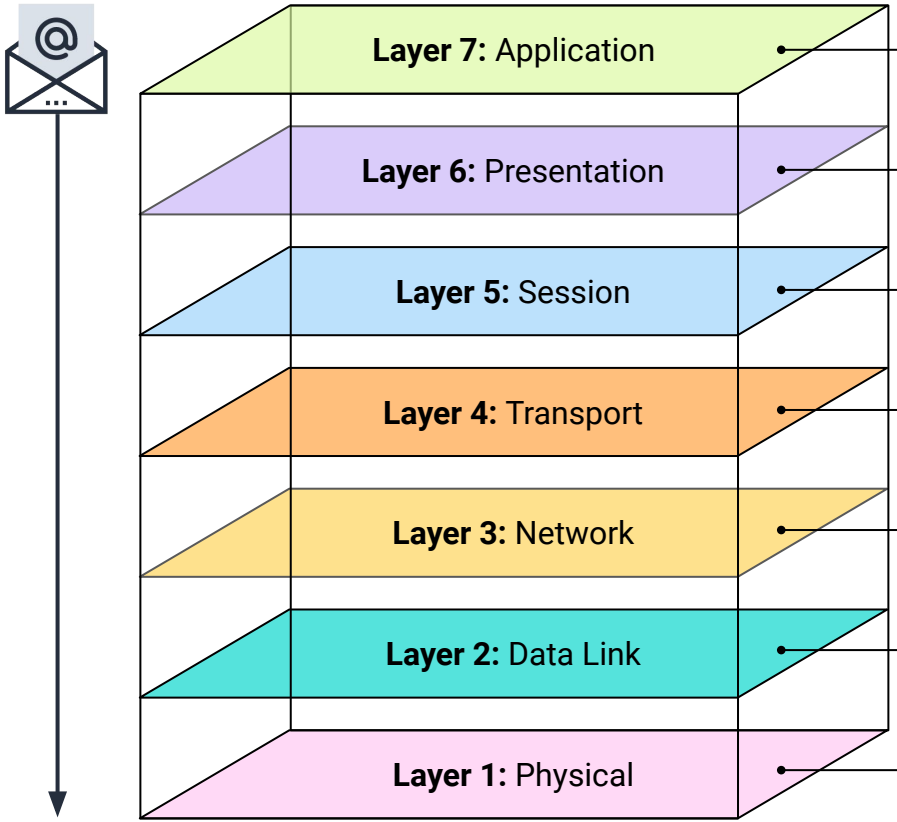
At the **Transport layer**, the receiving device analyzes the TCP header to determine the destination port intended to further process the data.

At the **Network layer**, the receiving device analyzes the IP header to determine if it contains the right IP address. If so, that header is removed and the data moves on to the next layer.

At the **Data Link**, the receiving device analyzes the MAC header to determine it matches the recipient's MAC address, therefore confirming it's arrived at the correct destination. If so, the device removes the header and moves to the next layer.

In summary, the TCP, IP, and MAC headers are decapsulated from the data.

An Email Moving through the OSI Layers



The user types an email in Microsoft Outlook.

The email text is converted from plain text into a version the receiving server can understand.

A session with the receiving mail server is initiated.

Since email uses SMTP, it assigns the SMTP destination port and initiates a handshake with the mail server.

The mail server destination IP is added.

The MAC address of the router is added so the email can be sent outside of the local network.

The digital email is converted into a signal to be transmitted over a physical cable.

OSI in a Security Context

The OSI model:	Example:
Helps us more easily understand new protocols.	If you find out that NetBIOS is a Layer 5 protocol, you immediately know that it's involved in managing user sessions, even if you've never heard of NetBIOS before.
Helps determine where problems in the network are occurring, even if we don't have full knowledge of the issue.	If you realize you're having a Layer 3 issue, you know you should start investigating your routers, even if you don't know exactly what the problem is.
Makes it easier to communicate where a security attack has occurred and what should be done.	If you know a SQL injection attack is occurring, you can explain to your management that you need a Layer 7 web application firewall to identify and mitigate the attack.

Activity: OSI Layers

In this activity, you will continue to play the role Security Analysts at Acme Corp.

You will be analyzing 10 suspicious network-related activities that have recently occurred at Acme Corp.

Your task is to document at which OSI layer each of these situations occurred.

Suggested Time:

15 Minutes

Time's Up! Let's Review.

Questions?


Recap

Recap

Let’s review network packets.

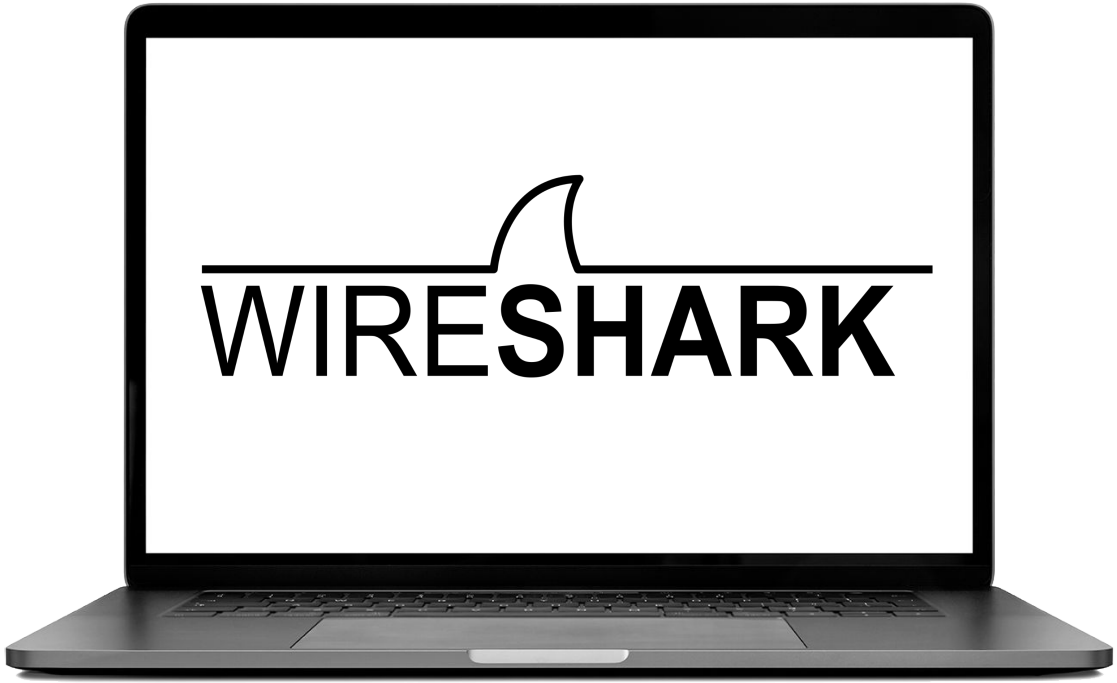
- Networks communicate with sequences of binary data called **packets**.
- Each packet contains fields such as the address of its origin, the address of its destination, and the information related packets being sent.
- These work a lot like the post office, except billions of packets are transferred each day, and most packets take less than a few seconds to reach their destination.
- Communicating over a network is not entirely safe, as these packets can be intercepted and analyzed by other users on the network.
- Cybersecurity professionals need to be able to see who's on a network and what they're doing. In other words, we have to be spies...at least a little bit.

Introduction to Wireshark

Introduction to Wireshark

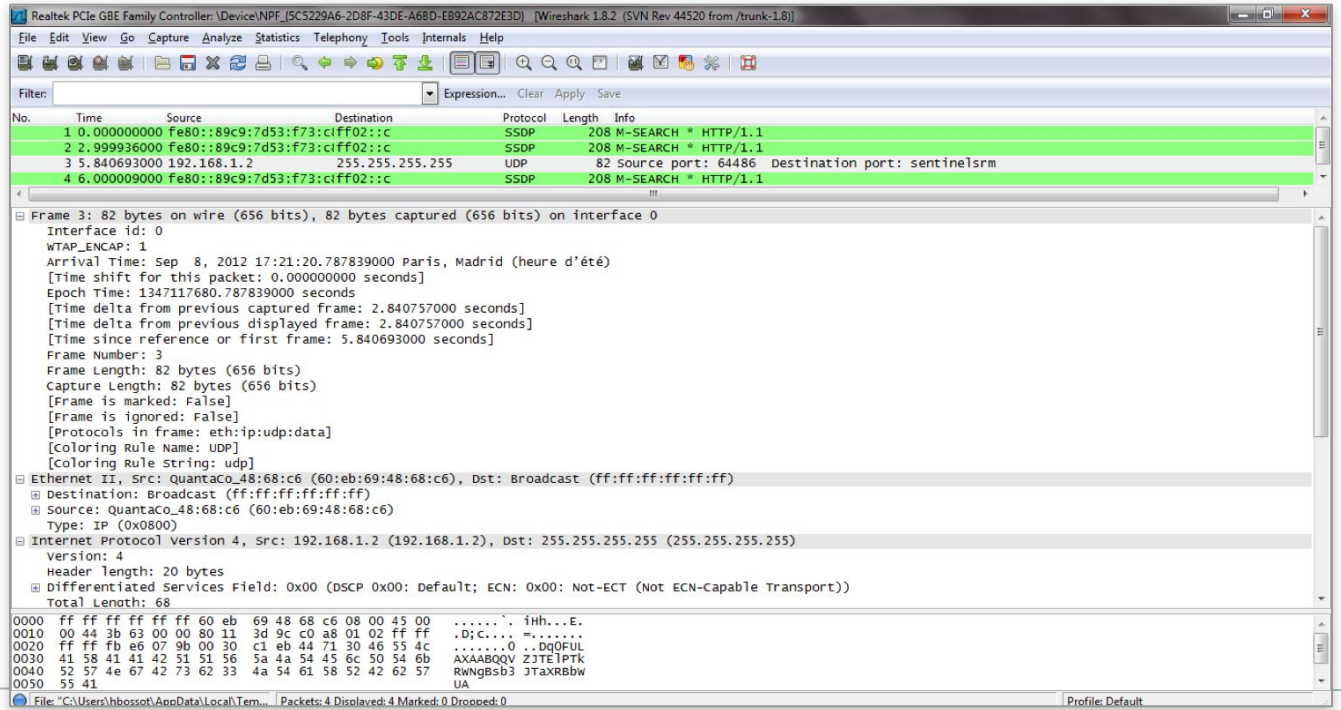
Wireshark is a tool that allows us to monitor real-time communications across a network, and the activities of the devices connected to it.

- Wireshark does this analysis by inspecting individual packets.
- Multiple packets collected into a file by Wireshark are called a **packet capture**.
- These have file extensions such as .cap, .pcap, and .pcapng.



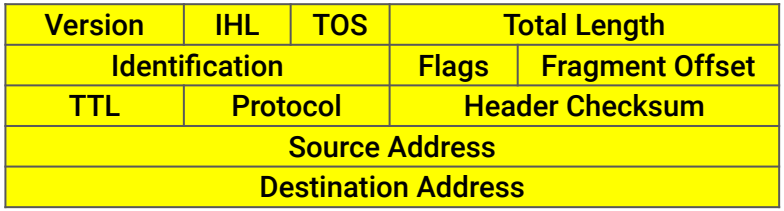
Capturing Packets

In these packet captures, Wireshark collects and analyzes the kinds of websites and webpages individuals on the network are viewing, as well as the type of communication occurring.



In the following slides, we'll break down the header of a packet capture.

Examine IPv4 Packet Header

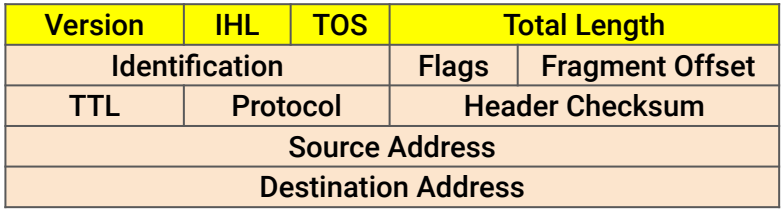


Sample Packet Capture

```
Internet Protocol Version 4, Src: host.docker.internatl (192.168.1.26), Dst: e7313.g.akamaie
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Fiels: 0×00 (DSCP: SD0, ECN: Not-ECT)
  Total Length: 40
  Identification: 0×b037 (45111)
> Flags: 0×40, Don't fragment
  Fragment Offset: 0
  Time to Live: 128
  Protocol: TCP (6)
  Header Checksum: 0×99ca [validation disabled]
  [Header checksum status: Unverified]
  Source Address: host.docker. Internal (192.168.1.26)
  Destination Address: e7313.g.akamaiedge.net (23.202.215.65)
```

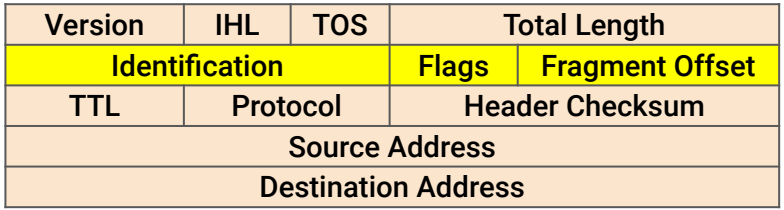
The **header** helps to identify details about what is going on with the packet.

Examine IPv4 Packet Header



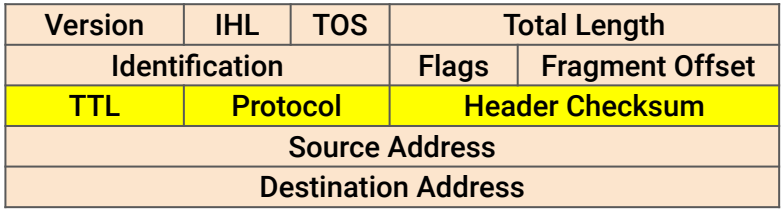

```
Internet Protocol Version 4, Src: host.docker.internatl (192.168.1.26), Dst: e7313.g.akamaie
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Fiels: 0×00 (DSCP: SD0, ECN: Not-ECT)
  Total Length: 40
  Identification: 0×b037 (45111)
> Flags: 0×40, Don't fragment
  Fragment Offset: 0
  Time to Live: 128
  Protocol: TCP (6)
  Header Checksum: 0×99ca [validation disabled]
  [Header checksum status: Unverified]
  Source Address: host.docker. Internal (192.168.1.26)
  Destination Address: e7313.g.akamaiedge.net (23.202.215.65)
```

Examine IPv4 Packet Header



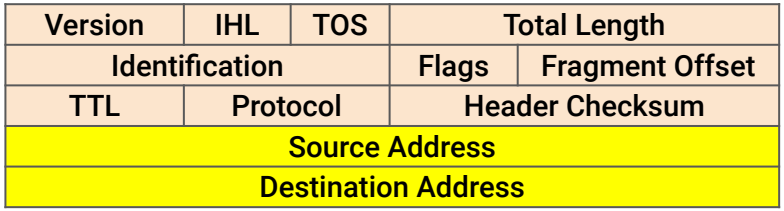

```
Internet Protocol Version 4, Src: host.docker.internatl (192.168.1.26), Dst: e7313.g.akamaie
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Fiels: 0×00 (DSCP: SD0, ECN: Not-ECT)
  Total Length: 40
  Identification: 0×b037 (45111)
> Flags: 0×40, Don't fragment
  Fragment Offset: 0
  Time to Live: 128
  Protocol: TCP (6)
  Header Checksum: 0×99ca [validation disabled]
  [Header checksum status: Unverified]
  Source Address: host.docker. Internal (192.168.1.26)
  Destination Address: e7313.g.akamaiedge.net (23.202.215.65)
```

Examine IPv4 Packet Header




```
Internet Protocol Version 4, Src: host.docker.internatl (192.168.1.26), Dst: e7313.g.akamaie
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Fiels: 0×00 (DSCP: SD0, ECN: Not-ECT)
  Total Length: 40
  Identification: 0×b037 (45111)
> Flags: 0×40, Don't fragment
  Fragment Offset: 0
  Time to Live: 128
  Protocol: TCP (6)
  Header Checksum: 0×99ca [validation disabled]
  [Header checksum status: Unverified]
  Source Address: host.docker. Internal (192.168.1.26)
  Destination Address: e7313.g.akamaiedge.net (23.202.215.65)
```

Examine IPv4 Packet Header




```
Internet Protocol Version 4, Src: host.docker.internatl (192.168.1.26), Dst: e7313.g.akamaie
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Fiels: 0×00 (DSCP: SD0, ECN: Not-ECT)
  Total Length: 40
  Identification: 0×b037 (45111)
> Flags: 0×40, Don't fragment
  Fragment Offset: 0
  Time to Live: 128
  Protocol: TCP (6)
  Header Checksum: 0×99ca [validation disabled]
  [Header checksum status: Unverified]
  Source Address: host.docker. Internal (192.168.1.26)
  Destination Address: e7313.g.akamaiedge.net (23.202.215.65)
```

Security professionals will often analyze network logs in order to research security-related issues.

Wireshark Professional Context

For example:

- Your manager has tasked you with analyzing web traffic to determine which source ports your system is using for HTTP requests.
- They want to make sure these aren't being blocked by your firewall.
- You've been provided a capture of the logs they want you to analyze.

You could look at the captured network logs using the command line.

Instructor Demonstration

Wireshark

Activity: Capturing Packets

In this activity, you will continue to play the role of a security analyst at Acme Corp.

You will configure your Wireshark application with the five requested configuration settings provided by your manager.

Suggested Time:

15 Minutes

Time's Up! Let's Review.

Questions?

Analyzing HTTP Traffic Setup

In the next demonstration, we will analyze HTTP web traffic with the following scenario:

Your manager wants you to make sure a new employee, Michael, is in fact working hard on his first day of work.

You could ask Michael if you could view his browser history, but, he could have cleared his browser history, or used more than one browser.

Fortunately, the networking team has a packet capture of Michael's web traffic, and you can use Wireshark to easily analyze the following:

- What websites has Michael visited?
- Were any communications sent from these websites?

Instructor Demonstration

Analyzing HTTP Web Traffic

Activity: Analyzing HTTP Data

In this activity, you will analyze web traffic to determine if Sally Stealer is a spy for your rival company, WidgetCorp.

You will also inspect the logs to see if Sally is sending any communications to WidgetCorp.

Suggested Time:

15 Minutes

Time's Up! Let's Review.

Questions?

Questions?


The End