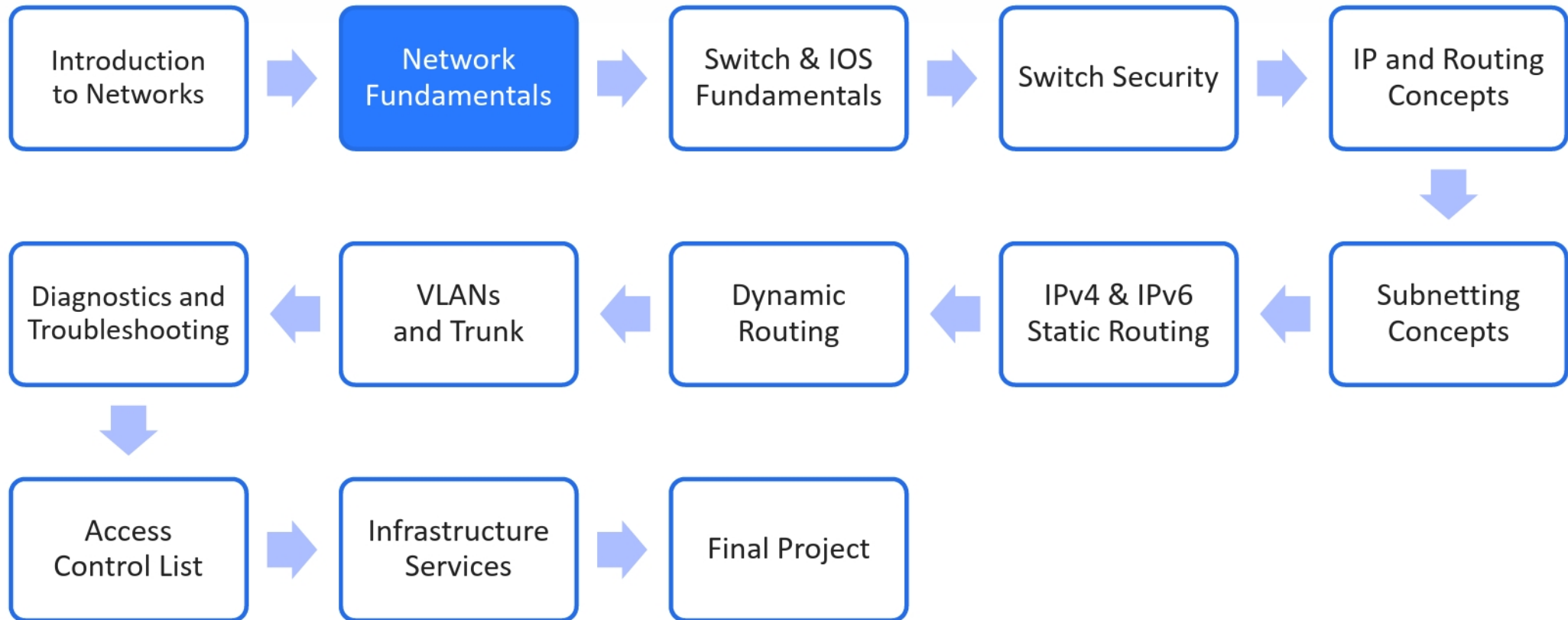


Cybersecurity Professional Program

Network Fundamentals

Computer Networking

Computer Networking Course Path



Overview

Lesson Objective

This module introduces the network components that together enable communication between different devices and services.

It also covers network architecture models and their layers.



Network Protocols and Models

OSI Model

TCP/IP Model

Transport Protocols and Logical Ports

Introduction to Wireshark Addresses

ARP and Encapsulation

Network Fundamentals

NETWORK PROTOCOLS AND MODELS

Client-Server

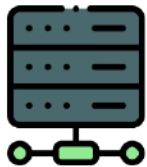
Client



A client represents the endpoint device (not necessarily a PC) that requested an operation.



Server



A server represents the endpoint device that receives a request and performs the requested operation.

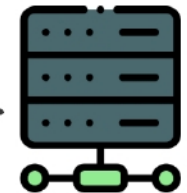


Web Browsing



Dear Server 18.77.22.44,
can you please send my
Facebook home page.

Sure! 10.2.2.7, here you
go, enjoy!



Client-Server

Client-Server Relationship



The Client-Server model is a distributed application structure that manages tasks and workloads by sharing them among providers of resources and services (servers), and delivers responses to service requesters (clients).



An alternative method is to use broadcast or multicast channels to distribute information.

Common Network Protocols

HTTP



HTTP is an unsecure protocol used to transfer information over the web.

Due to the fact that HTTP traffic is not encrypted, it can be sniffed on the network.

For example, if credit card information is entered in an HTTP website, it can be captured in simple text format.

HTTPS



HTTPS is the encrypted version of HTTP.

It is used for secure communication and is widely used on the Internet.

Communication is encrypted using TLS (Transport Layer Security) or SSL (Secure Socket Layer).

DNS

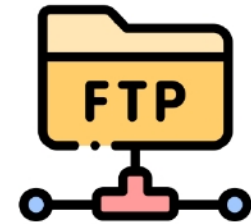


Translates domain names to IP addresses according to predefined databases.

The DNS protocol is used for address resolution in a network.

For example, when a URL is entered in a browser, the DNS protocol translates the domain that was entered to an IP address.

FTP



File Transfer protocol is used to transfer files over the network.

For example, the FTP service can be used to transfer files to another device over the network.

Common Network Protocols

DHCP



The DHCP service is responsible for assigning an IP address to each device on a network configured to use DHCP, such as PCs, Routers, Firewalls, Servers, etc.

For example, a PC configured with DHCP will automatically be assigned an IP address upon startup.

SMTP



Simple Mail Transfer Protocol is the standard protocol for mail transfer over the Internet.

For example, when an Email is sent over the network, the SMTP protocol manages the operation.

SSH



Secure SHell is a secure command line protocol that allows the user to run remote commands on a remote machine.

Any data that passes through SSH is encrypted.

For example, when a remote connection is made to a device, the connection is made using SSH on port 22.

ICMP



Internet Control Message Protocol is used by network devices to generate error messages when IP packets are not able to reach their destinations.

Network Models

Network Model Concept



The idea behind creating a networking model was to set a standard for the development and usage of network components.

Standards allow devices to communicate without requiring continuous configuration in different device platforms.

In addition, dividing the networking process into smaller sub-processes helps debug issues and pinpoint solutions.



TCP/IP and OSI were developed around the same time, with the aim of solving the same problem.

OSI Vs. TCP/IP

Today, OSI and TCP/IP are very similar due to the way they evolved over time. OSI is often used as an educational model, while TCP/IP is used more as a practical model.

OSI



OSI is a network model in which each functionality of the network is represented by a layer. There are 7 layers, in which the functionality of each is dependent on the one below it. OSI was developed by the International Organization of Standardization.



TCP/IP



The first TCP/IP specification was submitted several years after the OSI model. The original TCP/IP model consisted of 4 layers, but was later updated to the current 5 layers. However, the 4-layer version is still more commonly used. TCP/IP is used for network analysis, due to its lesser involvement with the top 3 layers.



Network Fundamentals

OSI MODEL

Open System Interconnection

OSI Model



- OSI defines a networking framework to implement protocols in layers.
- OSI conceptually divides computer network architecture into a logical seven-layer progression.
- Lower layers deal with electrical signals, chunks of binary data, and routing data across networks.
- Higher levels cover network requests and responses, data representation, and network protocols, from a user's viewpoint.



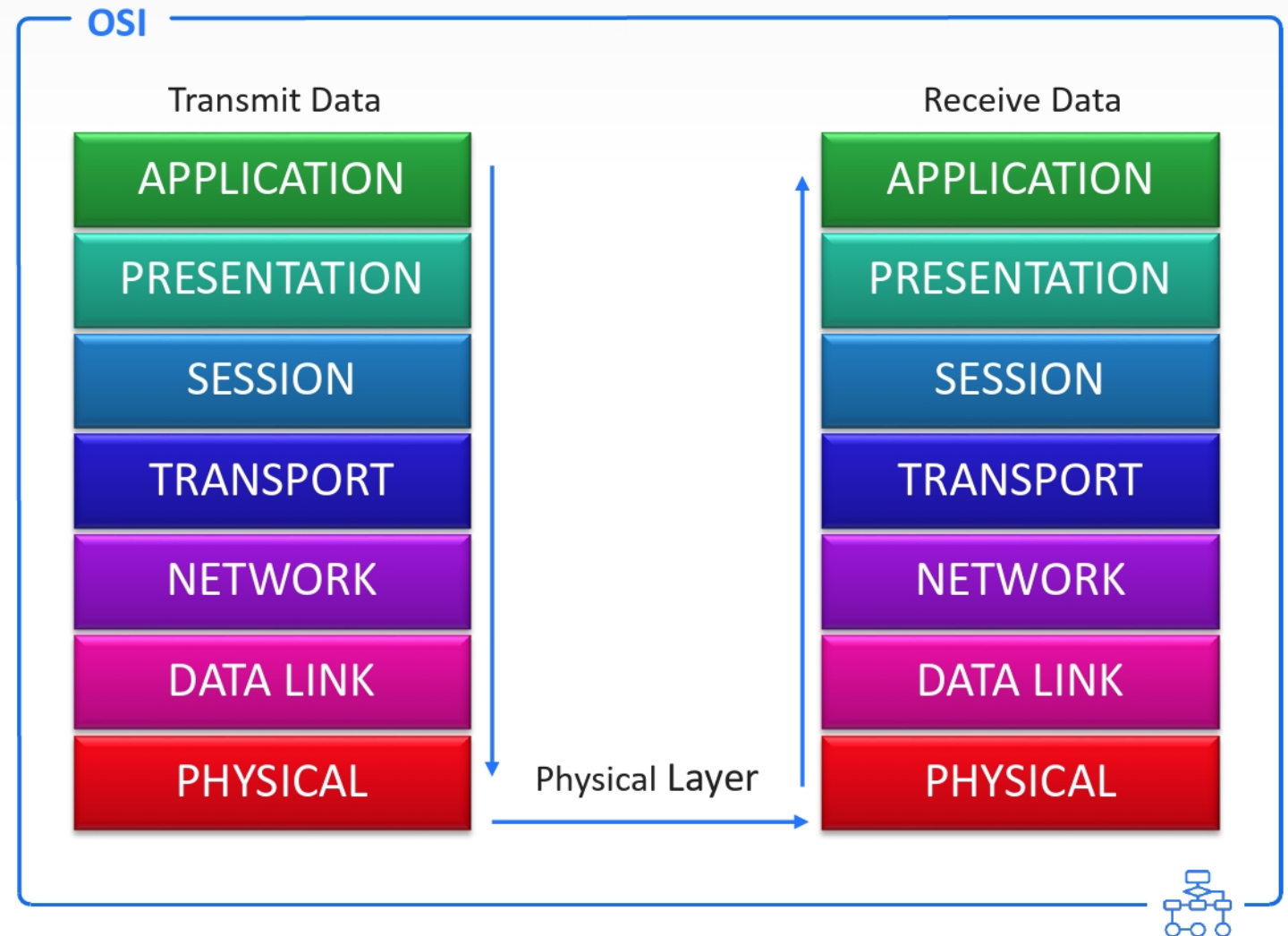
OSI was developed by the International Organization for Standardization (ISO) in 1974.

OSI Model

The OSI model is divided into 7 layers:
Application, Presentation, Session,
Transport, Network, Datalink, Physical.

When network traffic is generated, it
is assembled (encapsulated) from the
top layer to the bottom layer.

When received, traffic goes through
the model in the reverse direction:
from bottom to top (decapsulated).



Upper Layers

Layer 7 - Application



Users interact directly with applications that operate at Layer 7. Examples of Layer 7 applications include web browsers such as Google Chrome, Firefox, and Safari, and other applications, such as SSH and FTP.

Layer 6 - Presentation



The Presentation Layer prepares or translates data from application format to network format, and vice versa. An example of the Presentation Layer operation is the encryption and decryption process.

Layer 5 - Session



The session layer is responsible for creating a session between two devices. It is also responsible for session checkpoints and recovery.

Upper Layers

APPLICATION

PRESENTATION

SESSION



Transport Layer

Layer 4 - Transport



The Transport Layer handles coordination of data transfer between end systems and hosts, including how much data to send, at what rate, destination, etc. Data in this layer is organized into segments.

The transport layer determines if the connection should be instantiated, how to verify data integrity, how to recover from a connection error, and if retransmission is necessary.

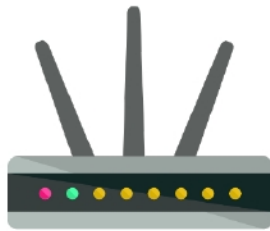
The protocols in this layer provide flow control, multiplexing, and reliability.



TCP and UDP are discussed in more detail in later slides.

Network Layer

Layer 3 - Network



The Network Layer organizes data into packets called IP datagrams that contain logical source and destination addresses.

Routers operate in this layer.

The Network Layer's job is to allow hosts to send packets across multiple paths to any other network, and deliver them to their destinations, via routers operating on this layer.



Diagnostic tools, such as ping and tracert operate in this layer.

Lower Layers

Layer 2 – Data Link



The Data Link Layer provides node-to-node data transfer (between two directly connected nodes) and can provide the means to detect errors that may occur in the Physical Layer. Switches operate in this layer. Data in this layer is organized into frames.



Layer 1 -Physical



The Physical Layer defines the details of how data is physically sent through the network. This layer determines how bits are electrically or optically transferred by hardware devices that interface directly with a network medium, such as coaxial cable, optical fiber, or twisted-pair copper wire. Hubs operate in this layer.



The Data Link Layer includes two sub-layers: MAC (Media Access Control) and LLC (Logical Link Control).

Protocol to Layer

Each network protocol is assigned to a particular OSI layer.

Common Protocols in Each Layer

LAYER	PROTOCOLS
Application	DNS, DHCP, SSH, HTTP\S, FTP, SNMP, SMTP, POP3
Presentation	Encryption, Encoding, SSL, ASCII, EBCDIC, TIFF, GIF, PICT, JPEG, MPEG, MIDI
Session	NFS, NetBios names, RPC, SQL
Transport	TCP, UDP, RTP, SCTP
Network	IPv4, IPv6, ICMPv4, ICMPv6, IPX
Data Link	Ethernet, PPP, FDDI, ATM, IEEE 802.5/ 802.2, HDLC, Frame Relay
Physical	Ethernet (IEEE802.3), Wi-Fi (IEEE 802.11), FDDI, B8ZS, V.35, V.24, RJ45



Memorize the Layers

Memory Tips



- All People Seem To Need Data Processing
- Please Do Not Throw Sausage Pizza Away
- Please Do Not Touch Steve's Pet Alligator



Data Encapsulation

What is Encapsulation?



When a message is sent from source to destination, it must have a specific format or structure. Message formats depend on the type of message and the channel used to deliver it.

The opposite process of encapsulation is called decapsulation. This is when the host is on the receiving side.



The same process occurs in OSI and TCP/IP using the same protocols.

Data Encapsulation

Encapsulation



For example, to send mail, we place a message in an envelope – encapsulation. The envelope is written in a format the mail service accepts and is sent to the destination. The receiving side opens the envelope and takes out the message – decapsulation.



The envelope is similar to a network Frame.

Encapsulation Process

As an example, we can focus on the process of sending data.
The first step in the process is ... YOU!

When you click on a link:

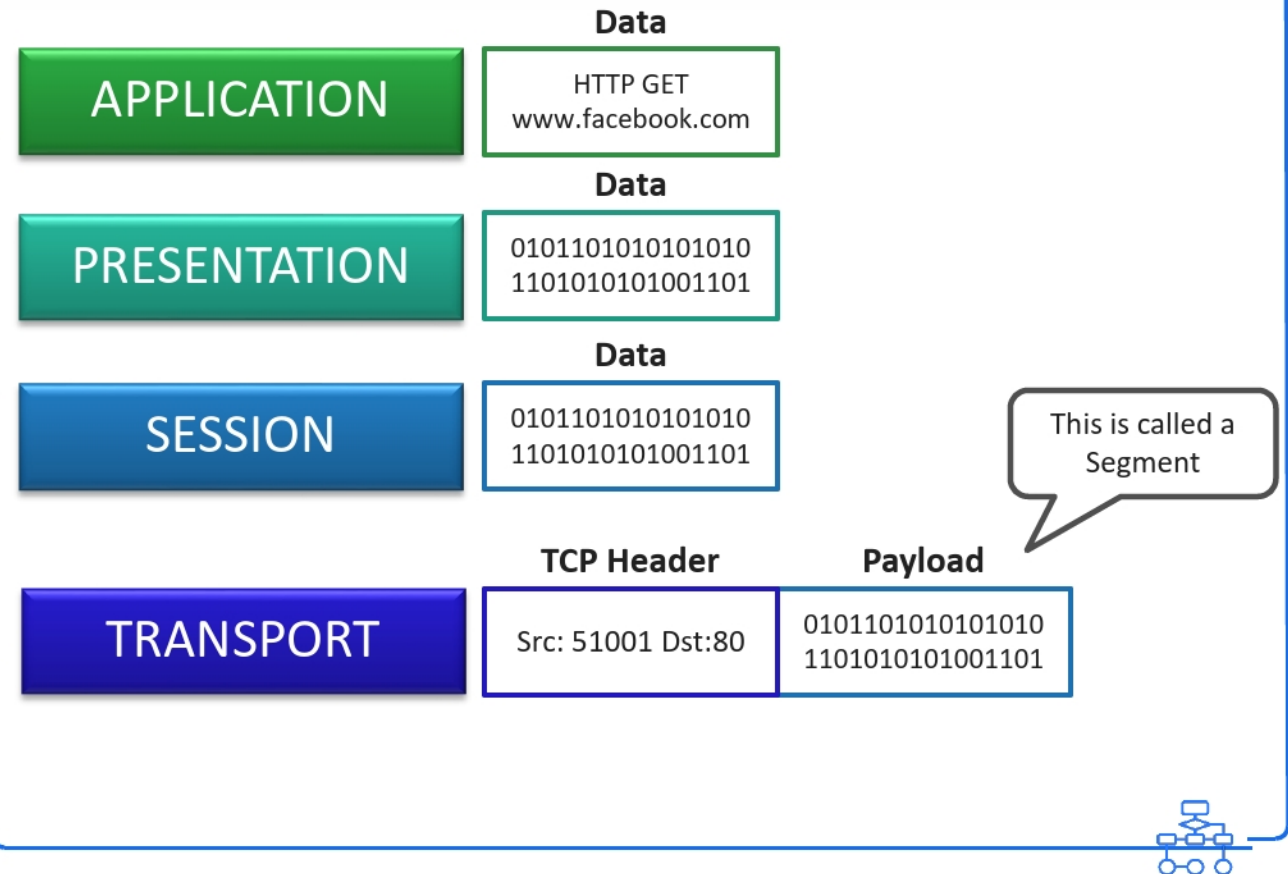
The **Application** layer creates an HTTP header requesting the web page from the server.

The **Presentation** layer encodes, encrypts, or compresses the data traveling down from the Application layer, if necessary. For example, it may convert text to a binary string.

The **Session** layer determines the direction of the data: one-way or two-way, and whether the connection is secure or not. In addition, a session ID is assigned to the connection.

The **Transport** layer adds a header containing the **source and destination logical ports**, and other control information. This layer also divides data from upper layers into smaller pieces called **Segments**. Each segment is a maximum (MSS) of 1460 bytes.

Encapsulation in Layers 4-7



Encapsulation Process

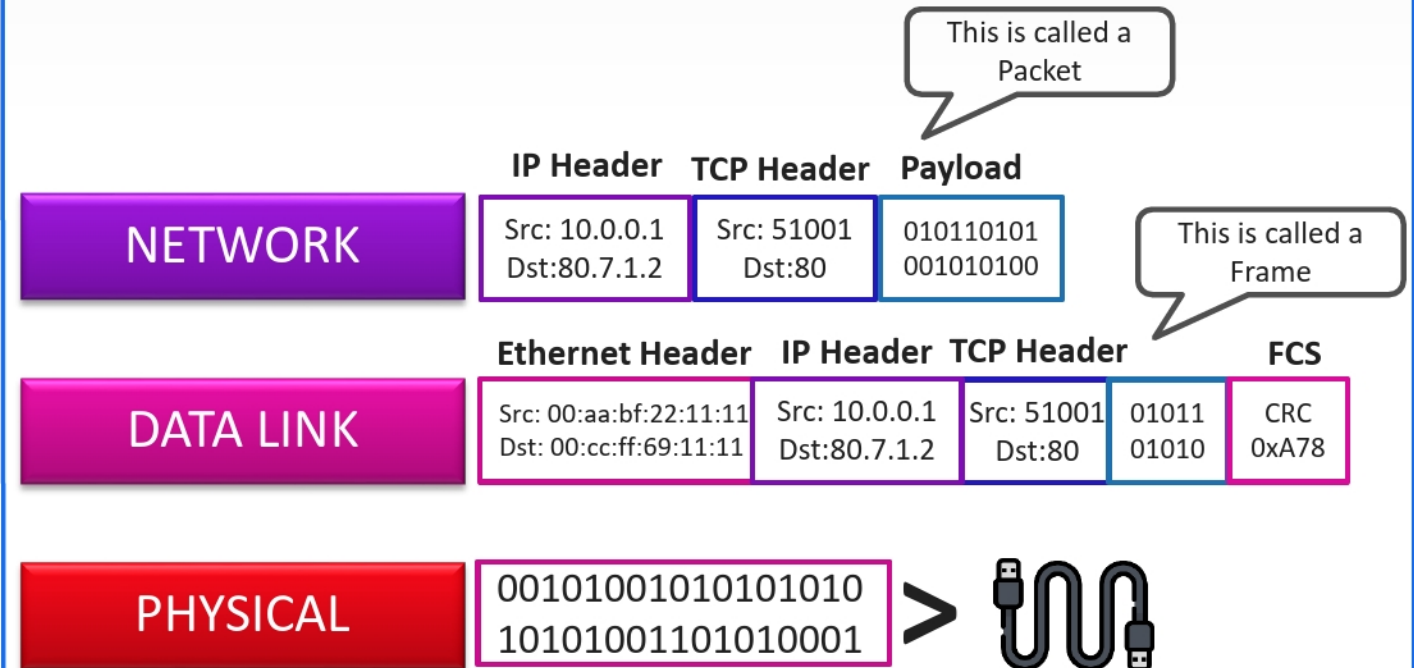
The **Network** Layer adds a header to the segment with the **IP source and destination** addresses, and other control information (such as TTL) . During this step, the segment is encapsulated in a packet.

Packets are limited to a maximum size (MTU – Maximum Transmission Unit) of **1500 bytes**.

The **Data Layer** adds a Header and Trailer to the **packet**. The header contains the **physical source and destination addresses**. The trailer, also known as FCS, is for frame error checking only.

The **Physical** layer accurately transmits the **Frames** along the physical medium. For example, it sends electric pulses over an Ethernet cable, or wireless radio waves.

Encapsulation in Layers 1-3

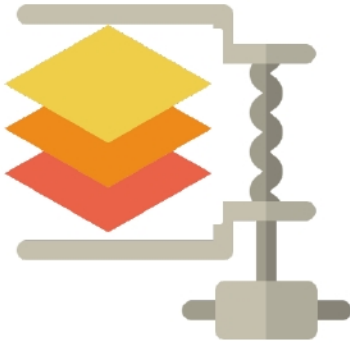


Network Fundamentals

TCP/IP MODEL

Transmission Control Protocol/Internet Protocol

TCP/IP Model



This model is a suite of communication protocols used to interconnect network devices on the Internet.

The model describes how data is exchanged over the Internet, including how data should be divided into packets, addressed, transmitted, routed, and received by the destination.

TCP/IP was designed to make networks reliable and enable them to automatically recover from the failure of any device on the network.



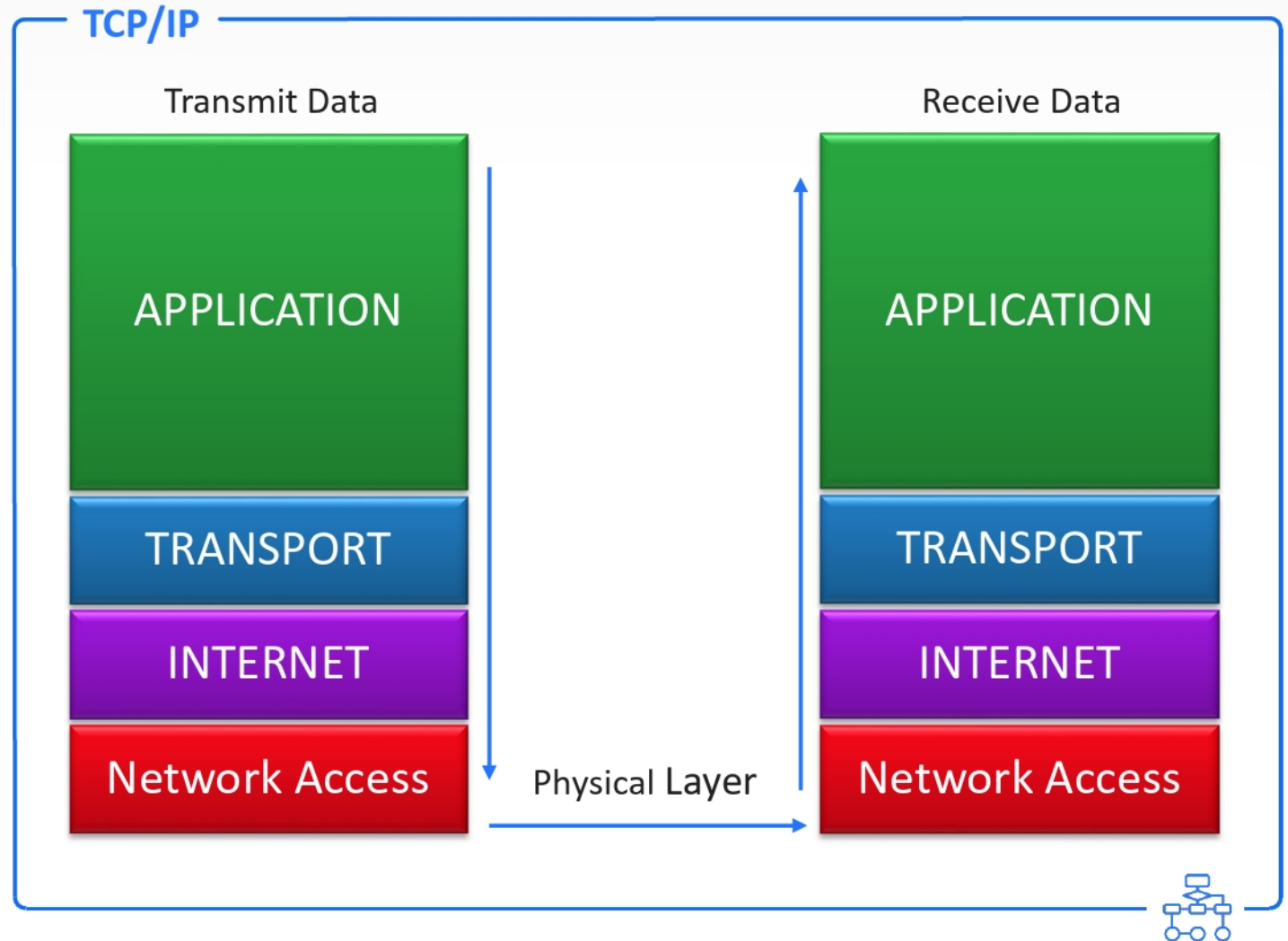
TCP/IP was developed by DARPA in the US Department of Defense, in 1982.

TCP/IP Layers

Layers in the TCP/IP model are similar to those in the OSI, except for the top three layers.

TCP/IP uses the client-server model of communication in which a user or machine is given access to a service by another computer on the network.

Collectively, the TCP/IP suite of protocols is classified as "Stateless", which means each client request is considered new, and is unrelated to previous requests.



Application Layer

Layer 4



The Application Layer defines TCP/IP application protocols and how host programs interface with Transport Layer services to use the network.

The Application Layer includes all higher-level protocols, such as DNS, HTTP, Telnet, SSH, FTP, TFTP, SNMP, SMTP, DHCP, X Windows, RDP, and others.



The TCP/IP top layer is equivalent to the combined three top layers of OSI.

Transport Layer

Layer 3



The Transport Layer handles coordination of data transfer between end systems and hosts, including how much data to send, at what rate, destinations, etc.

The transport layer determines if the connection should be instantiated, how to verify data integrity, how to recover from a connection error, and if retransmission is necessary. Data in this layer is organized into segments.

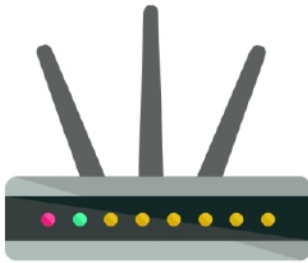
The protocols in this layer provide flow control, multiplexing, and reliability.



The Transport layer in TCP/IP corresponds to the Transport Layer in OSI.

Internet Layer

Layer 2



The Internet Layer organizes data into packets called IP datagrams that contain logical source and destination addresses.

Routers operate in this layer.

The Internet Layer's job is to allow hosts to send packets across multiple paths to any other network, and deliver them to their destinations, via routers operating on this layer.



The Internet layer in TCP/IP corresponds to the Network Layer in OSI.

Network Access

Layer 1



The Network Access Layer is the lowest layer of the TCP/IP protocol model.

Packets from the Internet layer are sent to the Network Access layer for delivery within the physical network. Data in this layer is organized into frames.

Protocols in this layer provide the means for the system to deliver data to other devices in a directly connected network, using physical addresses (MACs).

The protocols and standards defined for this layer dictate how data is transmitted on the media (Ethernet cable, wireless connection, others).



The Access Layer of the TCP/IP model corresponds with the Data Link and Physical layers of OSI.

Network Fundamentals

TRANSPORT PROTOCOLS AND LOGICAL PORTS

Transport Protocols

TCP & UDP



TCP and UDP are protocols that are used to transmit data over the network.

TCP and UDP operate in Layer 4 (Transport Layer).



There are other Transport Layer protocols, such as RTP and SCTP, but they are beyond the scope of this lesson.

TCP

Transmission Control Protocol



TCP is more trusted and commonly used than UDP.

It verifies the availability of all parties before initiating a connection, by performing a three-way handshake. It then marks packets and makes sure they are received in the correct order.



TCP is described in detail in RFC 793.

UDP

User Datagram Protocol



UDP is less strict about data integrity and is more speed-oriented.

UDP simply sends packets, without checking the data, even if nobody is ready to receive them.

This type of communication works better when streaming content, such as audio, where some packets may be skipped without affecting the entire stream.



UDP is described in detail in RFC 768.

TCP Vs UDP

TCP

Connection-oriented protocol.

Suitable for applications that require high reliability, where transmission time is less critical.

Rearranges data packets in the order specified.

Guarantees that transferred data remains intact. If a packet is corrupt upon arrival, it is resent.

Errors are detected via checksum.

TCP is relatively slow.



VS

UDP

A connectionless protocol.

Suitable for applications that need fast, efficient transmission. Also useful for servers that answer small queries from large numbers of clients.

Packets may arrive out of order, UDP does not rearrange them.

There is no guarantee that messages or packets that are sent will reach their destinations.

UDP does error-checking but simply discards erroneous packets. Error recovery is not attempted.

UDP is relatively fast.



TCP Flags

TCP uses flags to provide additional useful information or control the connection.

TCP Flags

Flag	Role
SYN	Synchronization - this is the first flag between two hosts, and is used to establish a 3-way handshake.
ACK	Acknowledged - acknowledges successful receipt by the host.
RST	Reset - Terminates a connection if a TCP failure occurs or a conversation is not permitted.
FIN	Finish - Requests connection termination.
PSH	Push - Immediately sends all buffered data.
URG	Urgent - Data with a URG segment is forwarded to the Applications Layer immediately.



Transport Protocols and Logical Ports

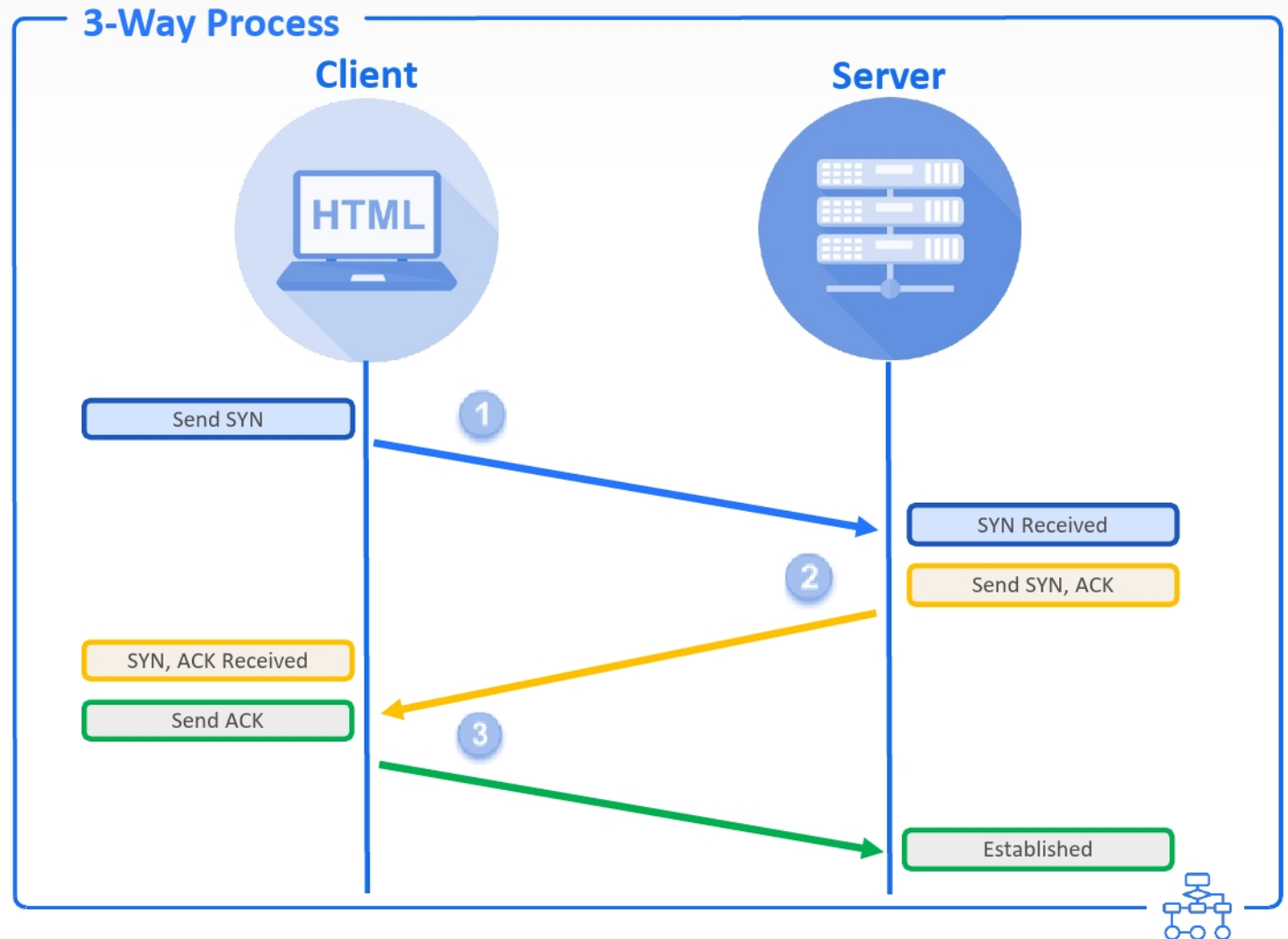
3-Way Handshake

A 3-way handshake is responsible for communication between network devices that use TCP.

The handshake is known as “3-way” because it includes 3 steps:

1. The client sends “SYN”.
2. The server responds with “SYN-ACK”.
3. The client finalizes with “ACK”.

When the process ends successfully, a connection is established between the client and the server, and data can immediately be transferred between them.



Session Termination

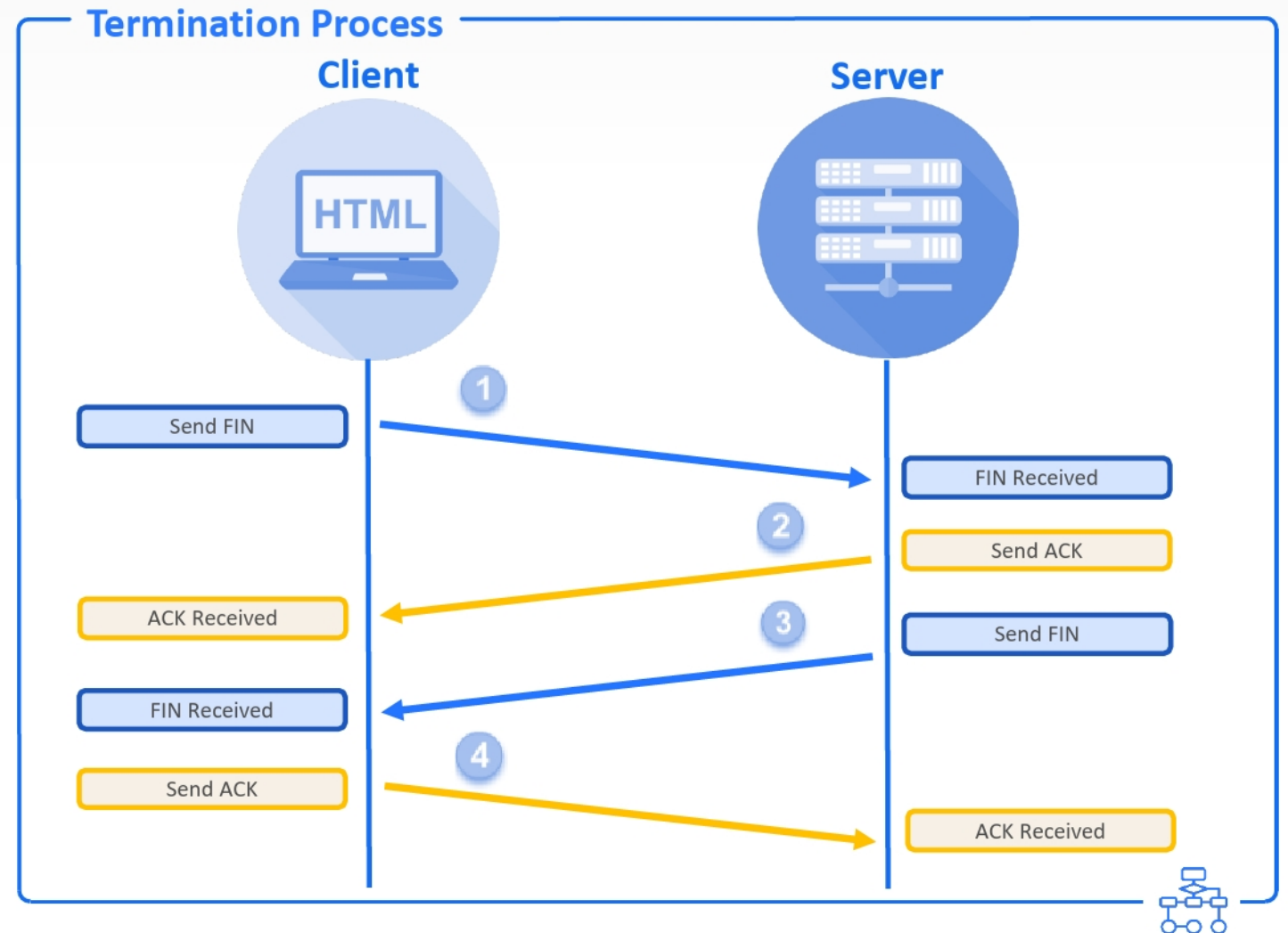
Session Termination is the last stage of a TCP connection. This is when communication ends and the session is closed.

Step 1 - One of the sides (client) requests to close the connection by sending a FIN flag to the other side (server).

Step 2 - The receiver of the FIN flag (server) returns an ACK flag, to approve the request.

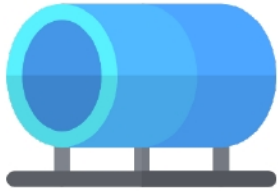
Step 3 - The server then sends a FIN flag to the client.

Step 4 - The client approves the FIN request by the server, and responds with an ACK flag. The connection is then terminated.



Logical Ports

What are Logical Ports?



In computer science, a logical port is a number assigned to a “logical” connection, which is used to represent a service or protocol on the end device.

An IP address can send packets to the destination host, but not to the destination service on that end device.

As an example, a street address can lead to the correct building, but it does not specify the apartment number within that building.



Logical Port Groups

There are 65,536 in TCP and 65,536 UDP ports, divided into three main groups, as follows:

Well-Known Ports

0-1023

Port numbers in the “well-known” port range are assigned to commonly used and essential protocols. The port numbers are permanent, and every network administrator must know them. We use these numbers all the time, without realizing it.

The range includes protocols such as HTTPS, DNS, DHCP, Email protocols, and others.



Registered Ports

1024 -
49151

IANA assigned these ports for IT companies that want to register them for proprietary applications. For example, Microsoft RDP 3389 and Dropbox 7600.



Logical Port Groups

Dynamic & Private Ports

49152 -
65535

This group covers the remaining logical ports that are available for any application. The port range is known as “client ports” and is usually used for logical client connection.

For example, a host connecting to a web server sends requests to port 80. This lets the web server know which service the client is requesting. However, the return traffic must be sent to a source port, which is chosen randomly. When the connection closes, the port can be reused for another connection, enabling the client to establish multiple connections and manage responses from multiple sources.



Common Protocol Ports

The following table lists common network protocols and their TCP and UDP ports.

Logical Port Numbers

Protocol	TCP\UDP	Logical Ports	Protocol	TCP\UDP	Logical Ports
HTTP	TCP	80	SSH	TCP	22
HTTPS	TCP	443	RDP	TCP	3389
DNS	UDP	53	SMB	TCP	445
DHCP	UDP	67, 68	FTP	TCP	20, 21
SMTP	TCP	25, 587	TFTP	UDP	69
POP3	TCP	110	NTP	UDP	123



Viewing Open Ports

Ports are used to receive and send data.

A web server will open Port 80 in listening mode if it intends to receive HTTP traffic.

A client will open a random port for each server it accesses when browsing the web.

To inspect open ports, use the command **netstat -ab**
a stands for all.
b stands for the names.

Netstat

```
C:\WINDOWS\system32>netstat -ab  
Active Connections
```

Proto	Local Address	Foreign Address	State
TCP	0.0.0.0:135 [svchost.exe]	Server_01:0	LISTENING
TCP	0.0.0.0:443 [vmware-hostd.exe]	Server_01:0	LISTENING
TCP	0.0.0.0:445	Server_01:0	LISTENING
Can not obtain ownership information			
TCP	0.0.0.0:902 [vmware-authd.exe]	Server_01:0	LISTENING
TCP	0.0.0.0:912 [vmware-authd.exe]	Server_01:0	LISTENING
TCP	0.0.0.0:1536 [svchost.exe]	Server_01:0	LISTENING
TCP	0.0.0.0:1538 [svchost.exe]	Server_01:0	LISTENING
TCP	0.0.0.0:1539	Server_01:0	LISTENING



Quiz Time

According to the materials learned in the previous part, answer the following questions as best as you can. You may use external materials, such as Internet searches.

Communication Roles

Number	Question
1	Briefly describe the operation or function of each layer. (For example, Layer 6 encodes, encrypts, and compresses the information)
2	List two to three main protocols that operate in each layer. (For example IP, ICMP-Network Layer)
3	List two protocols that use TCP, and two that use UDP.
4	Describe the TCP 3-Way Handshake.
5	List the logical ports of the following protocols: HTTP, HTTPS, DNS, FTP, SMTP.
6	How many layers do the TCP/IP and OSI models have, and what are their names? (Try to answer from memory)



Network Fundamentals

INTRODUCTION TO WIRESHARK

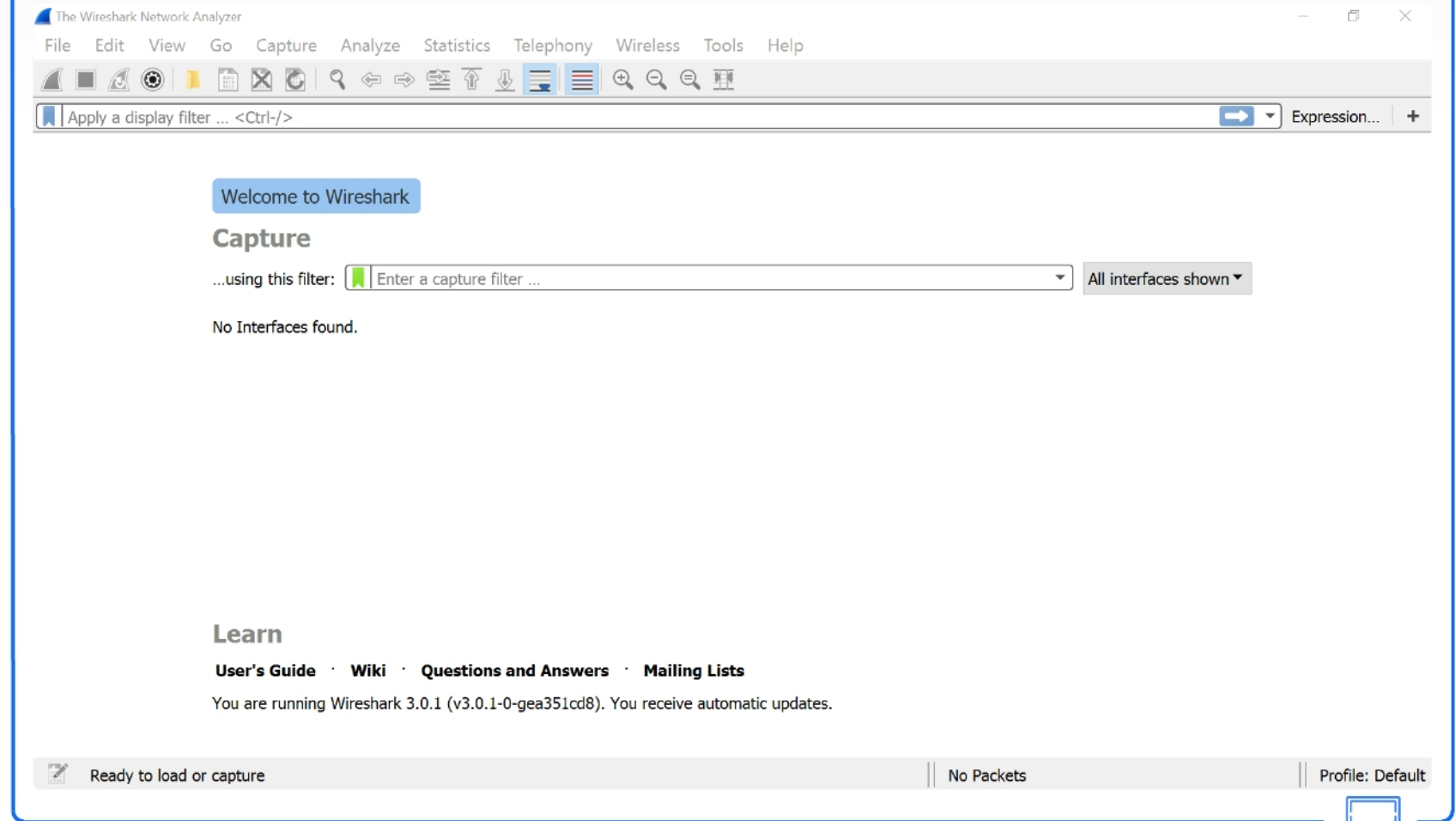
Introduction to Wireshark

Wireshark is a free application that enables the capture and viewing of data traveling back and forth on the network.

It is able to read the contents of each packet and apply different filters.

Wireshark is commonly used to troubleshoot network problems, and develop and test software.

Wireshark



Capturing Data

When invoked, Wireshark presents an initial screen with a list of available network interfaces.

Selecting an interface will start the capture of all traffic that passes through it.

Wireshark can also be used to open traffic captured by other tools, such as TCP dump.

Captured data can be exported into .pcap or .pcapng files.

Wireshark

The screenshot displays the Wireshark interface with a list of 12 captured packets. The packets are filtered by 'tcp'. The details pane shows the structure of the first packet (Frame 1), which is an Ethernet II frame containing an Internet Protocol Version 4 packet and a Transmission Control Protocol (TCP) segment.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.1.20.1	157.240.1...	TLSv1.2	85	Application Data
2	0.078309	157.240.1.53	10.1.20.1	TCP	60	443 → 8199 [ACK] Seq=1 Ack=32 Win=144 Len=0
3	0.131308	10.1.20.1	62.0.136....	TCP	55	[TCP segment of a reassembled PDU]
4	0.131571	62.0.136.237	10.1.20.1	TCP	66	8093 → 11632 [ACK] Seq=1 Ack=2 Win=254 Len=0 SLE=1 SRE=2
5	0.155242	157.240.1.53	10.1.20.1	TLSv1.2	92	Application Data
6	0.195710	10.1.20.1	157.240.1...	TCP	54	8199 → 443 [ACK] Seq=32 Ack=39 Win=1027 Len=0
7	1.131756	10.1.20.1	62.0.136....	TCP	55	[TCP Keep-Alive] 11632 → 8093 [ACK] Seq=1 Ack=1 Win=8212 L
8	1.132066	62.0.136.237	10.1.20.1	TCP	66	[TCP Keep-Alive ACK] 8093 → 11632 [ACK] Seq=1 Ack=2 Win=25
9	1.228919	146.185.219.33	10.1.20.1	TCP	60	80 → 49803 [ACK] Seq=1 Ack=1 Win=501 Len=0
10	1.228960	10.1.20.1	146.185.2...	TCP	54	[TCP ACKed unseen segment] 49803 → 80 [ACK] Seq=1 Ack=2 Wi
11	2.131958	10.1.20.1	62.0.136....	TCP	55	[TCP Keep-Alive] 11632 → 8093 [ACK] Seq=1 Ack=1 Win=8212 L
12	2.132228	62.0.136.237	10.1.20.1	TCP	66	[TCP Keep-Alive ACK] 8093 → 11632 [ACK] Seq=1 Ack=2 Win=25

Frame 1: 85 bytes on wire (680 bits), 85 bytes captured (680 bits) on interface 0
 Ethernet II, Src: HewlettP_3e:5b:9e (f4:39:09:3e:5b:9e), Dst: Fortinet_4e:8b:12 (04:d5:90:4e:8b:12)
 Internet Protocol Version 4, Src: 10.1.20.1, Dst: 157.240.1.53
 Transmission Control Protocol, Src Port: 8199, Dst Port: 443, Seq: 1, Ack: 1, Len: 31

0000 04 d5 90 4e 8b 12 f4 39 09 3e 5b 9e 08 00 45 00 ...N...9 >[...E
 0010 00 47 da 81 40 00 80 06 63 08 0a 01 14 01 9d f0 .G...@... c.....
 0020 01 35 20 07 01 bb 5a 42 67 7e a0 0c 09 b2 50 18 .5 ...ZB g~...P.
 0030 04 03 86 0e 00 00 17 03 03 00 1a ca 99 84 e4 00

Wireshark_Ethernet_20200721130008_a14160.pcapng | Packets: 522 · Displayed: 496 (95.0%) | Profile: Default

Wireshark - Main View

Packet List



Packet List is the main view of Wireshark. It lists captured packets in the order of their capture, and offers various search and filter options.



Packet Details



Packet Details is the part in the middle that displays the details of a specific packet. The details are divided into fields corresponding to TCP/IP layers, such as the Frame (Physical), Ethernet (Data Link), and IP (Network).



Packet Bytes



The bottom part of the view corresponds to the actual bytes of the selected packet, in hexadecimal format and ASCII encoding.



Introduction to Wireshark

ICMP Traffic

The Ping command includes a request and a reply.

Each ICMP packet is numbered and can be identified to find a corresponding response.

ICMP has only four fields since it is a Layer 3 protocol.

In Windows each ICMP echo (ping) packet includes all the basic symbols and numbers on the keyboard.

Wireshark

ICMP Sample.cap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> Expression...

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.42.0.157	172.42.0.140	ICMP	98	Echo (ping) request id=0xcda, seq=1/256, ttl=64
2	0.300934	172.42.0.140	172.42.0.157	ICMP	98	Echo (ping) reply id=0xcda, seq=1/256, ttl=64
3	1.000925	172.42.0.157	172.42.0.140	ICMP	98	Echo (ping) request id=0xcda, seq=2/512, ttl=64
4	1.298136	172.42.0.140	172.42.0.157	ICMP	98	Echo (ping) reply id=0xcda, seq=2/512, ttl=64

Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)

Ethernet II, Src: Vmware_c6:21:44 (00:0c:29:c6:21:44), Dst: SamsungE_83:45:3b (04:d6:aa:83:45:3b)

Internet Protocol Version 4, Src: 172.42.0.157, Dst: 172.42.0.140

Internet Control Message Protocol

Type: 8 (Echo (ping) request)

Code: 0

Checksum: 0x7d90 [correct]

[Checksum Status: Good]

Identifier (BE): 3290 (0xcda)

Identifier (LE): 55820 (0xda0c)

Sequence number (BE): 1 (0x0001)

Sequence number (LE): 256 (0x0100)

0000 04 d6 aa 83 45 3b 00 0c 29 c6 21 44 08 00 45 00 ...E;...)!D..E.

0010 00 54 8c 1a 40 00 40 01 55 11 ac 2a 00 9d ac 2a .T..@. @. U..*...*

0020 00 8c 08 00 7d 90 0c da 00 01 e9 53 bb 5c 00 00 ...}... ..S.\..

0030 00 00 04 11 06 00 00 00 00 00 10 11 12 13 14 15

0040 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 !"#\$\$%

0050 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 &'()*+,- ./012345

ICMP Sample.cap

Packets: 8 · Displayed: 8 (100.0%)

Profile: Default

DNS Traffic

DNS traffic includes multiple requests and replies.

Most of a search tree will not appear in a packet. Only the request and response.

DNS may try to automatically resolve both A and AAAA records corresponding to the IPv4 and IPv6 addresses respectively.

Wireshark

The image shows a Wireshark capture of DNS traffic. The packet list on the left shows six packets. The selected packet (No. 1) is a DNS Standard query from 172.42.0.151 to 172.42.0.138. The packet details on the right show the query for api.segment.io. The packet bytes at the bottom show the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.42.0.151	172.42.0.138	DNS	74	Standard query 0xd282 AAAA api.segment.io
2	0.010187	172.42.0.138	172.42.0.151	DNS	158	Standard query response 0xd282 AAAA api.segment.io SOA
3	5.906976	172.42.0.157	172.42.0.138	DNS	71	Standard query 0xb970 A [redacted]
4	5.994438	172.42.0.138	172.42.0.157	DNS	87	Standard query response 0xb970 A [redacted].com A 107.180.
5	5.995163	172.42.0.157	172.42.0.138	DNS	71	Standard query 0x0276 AAAA [redacted]
6	6.121930	172.42.0.138	172.42.0.157	DNS	142	Standard query response 0x0276 AAAA [redacted].com SOA ns1

Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
Ethernet II, Src: LcfcHefe_7a:4b:17 (e8:6a:64:7a:4b:17), Dst: D-LinkIn_d7:48:d0 (78:32:1b:d7:48:d0)
Internet Protocol Version 4, Src: 172.42.0.151, Dst: 172.42.0.138
User Datagram Protocol, Src Port: 62297, Dst Port: 53
Domain Name System (query)
Transaction ID: 0xd282
Flags: 0x0100 Standard query
Questions: 1
Answer RRs: 0
Authority RRs: 0
Additional RRs: 0

0000 78 32 1b d7 48 d0 e8 6a 64 7a 4b 17 08 00 45 00 x2..H..j dzK...E.
0010 00 3c 6f bb 00 00 80 11 71 80 ac 2a 00 97 ac 2a .<o.....q..*...*
0020 00 8a f3 59 00 35 00 28 b1 10 d2 82 01 00 00 01 ...Y.5.(.....
0030 00 00 00 00 00 00 03 61 70 69 07 73 65 67 6d 65a pi·segme
0040 6e 74 02 69 6f 00 00 1c 00 01 nt-io... ..



Introduction to Wireshark

HTTP Traffic

HTTP is an unsecure protocol used to transfer information over the web.

Since HTTP traffic is not encrypted, it can be sniffed on the network.

Wireshark

HTTP Sample.cap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http

No.	Time	Source	Destination	Protocol	Length	Info
16	14.384881	172.42.0.157	192.195.77.80	HTTP	779	POST /login-form/ HTTP/1.1 (application/x-www-form-urlencoded)
22	14.571270	192.195.77.80	172.42.0.157	HTTP	357	HTTP/1.1 200 OK (text/html)

Frame 16: 779 bytes on wire (6232 bits), 779 bytes captured (6232 bits)

- Ethernet II, Src: Vmware_c6:21:44 (00:0c:29:c6:21:44), Dst: D-LinkIn_d7:48:d0 (78:32:1b:d7:48:d0)
- Internet Protocol Version 4, Src: 172.42.0.157, Dst: 192.195.77.80
- Transmission Control Protocol, Src Port: 60834, Dst Port: 80, Seq: 1, Ack: 1, Len: 713
- Hypertext Transfer Protocol
- HTML Form URL Encoded: application/x-www-form-urlencoded
 - Form item: "username" = "lionk"
 - Form item: "password" = "Aa123456"
 - Form item: "submit" = "Login"

0000 78 32 1b d7 48 d0 00 0c 29 c6 21 44 08 00 45 00 x2..H...)!D..E.
0010 02 fd bf 2f 40 00 40 06 bd f0 ac 2a 00 9d c0 c3 .../@.@ ...*....
0020 4d 50 ed a2 00 50 dd 31 94 33 cb c3 de c6 80 18 MP...P-1 -3.....
0030 00 e5 bd ca 00 00 01 01 08 0a c7 af b0 43 d9 21C.!
0040 5f ed 50 4f 53 54 20 2f 6c 6f 67 69 6e 2d 66 6f _POST / login-fo
0050 72 6d 2f 20 48 54 54 50 2f 31 2e 31 0d 0a 48 6f rm/ HTTP /1.1..Ho
0060 73 74 3a 20 77 77 77 2e 64 65 6d 6f 2e 61 6d 69 st: www. demo.ami
0070 74 6a 61 6b 68 75 2e 63 6f 6d 0d 0a 55 73 65 72 tjakhu.c om..User
0080 2d 41 67 65 6e 74 3a 20 4d 6f 7a 69 6c 6c 61 2f -Agent: Mozilla/
0090 35 2e 30 20 28 58 31 31 3b 20 4c 69 6e 75 78 20 5.0 (X11 ; Linux
00a0 78 38 36 5f 36 34 3b 20 72 76 3a 36 30 2e 30 29 x86_64; rv:60.0)

HTTP Sample.cap

Packets: 31 · Displayed: 2 (6.5%)

Profile: Default

Filtering Traffic

Filters are logical conditions that are applied using **&&**, **||**, **==**, and **!=**. Brackets can also be used.



Wireshark Filters

Wireshark includes many display filters. The following table lists some useful ones.

Logical Port Groups

Filter	Data
Protocol Name	Find a match for packets using a specific protocol (i.e. icmp, arp, http, dns).
ip.addr == 10.0.0.3	Find a match for this address on the source and destination.
ip.dst != 10.0.0.2	Find all matches that are not this destination.
ip.src != 10.0.0.3	Find all matches that are not this source.
tcp.port == 25 && tcp.port == 80	Find packets that match both specified port numbers in TCP.
udp.port == 53 udp.port == 69	Find packets that match one of the specified port numbers in UDP.
udp.dstport == 25	Find packets that match a specific port number on the destination in UDP.



Following Traffic

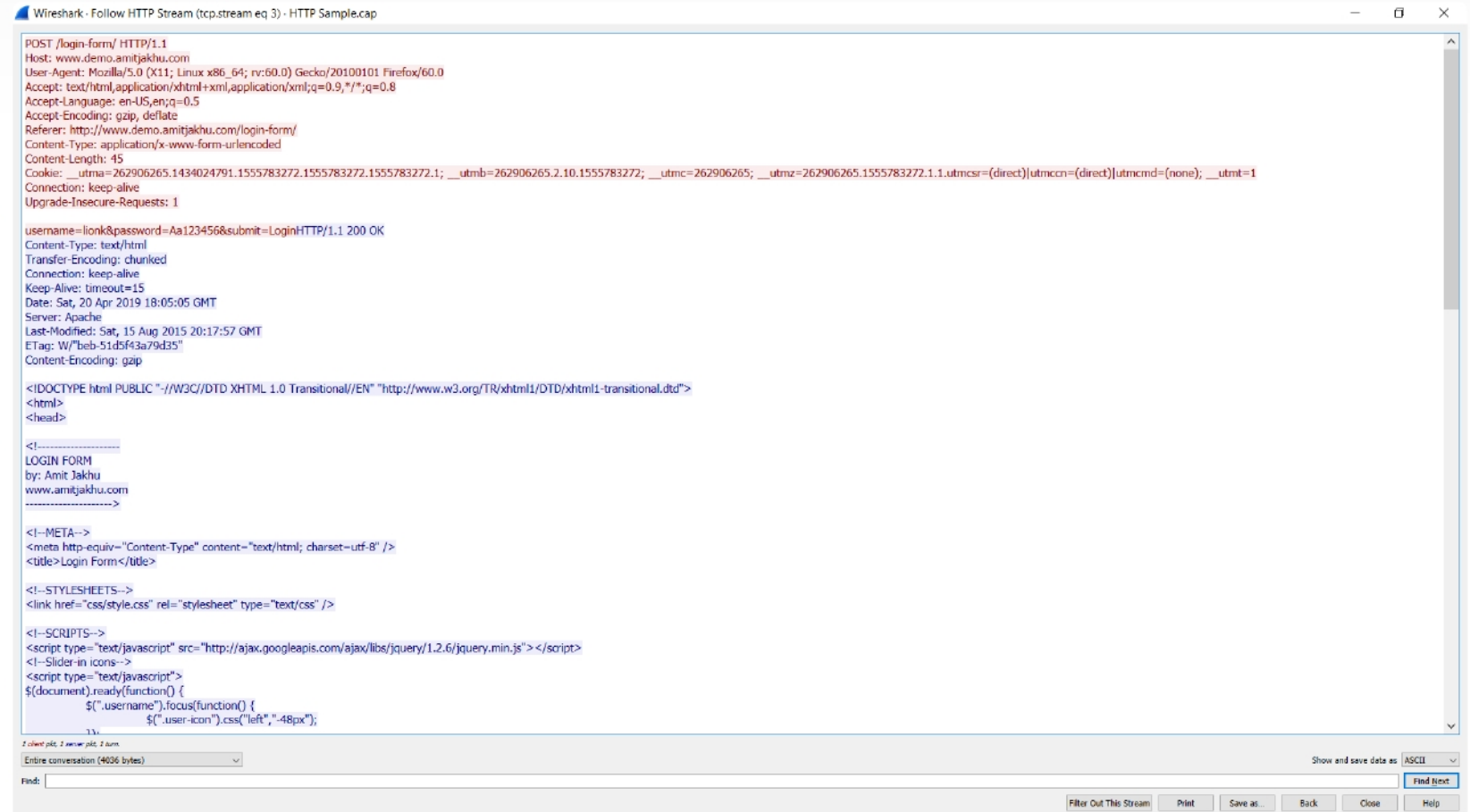
After capturing traffic and filtering, stream inspection can be done.

Streams are collections of packets from the same conversation.

Left-click a packet to choose the option to follow a stream, which can be TCP, UDP, or HTTP.

Unless traffic is encrypted, it should be readable.

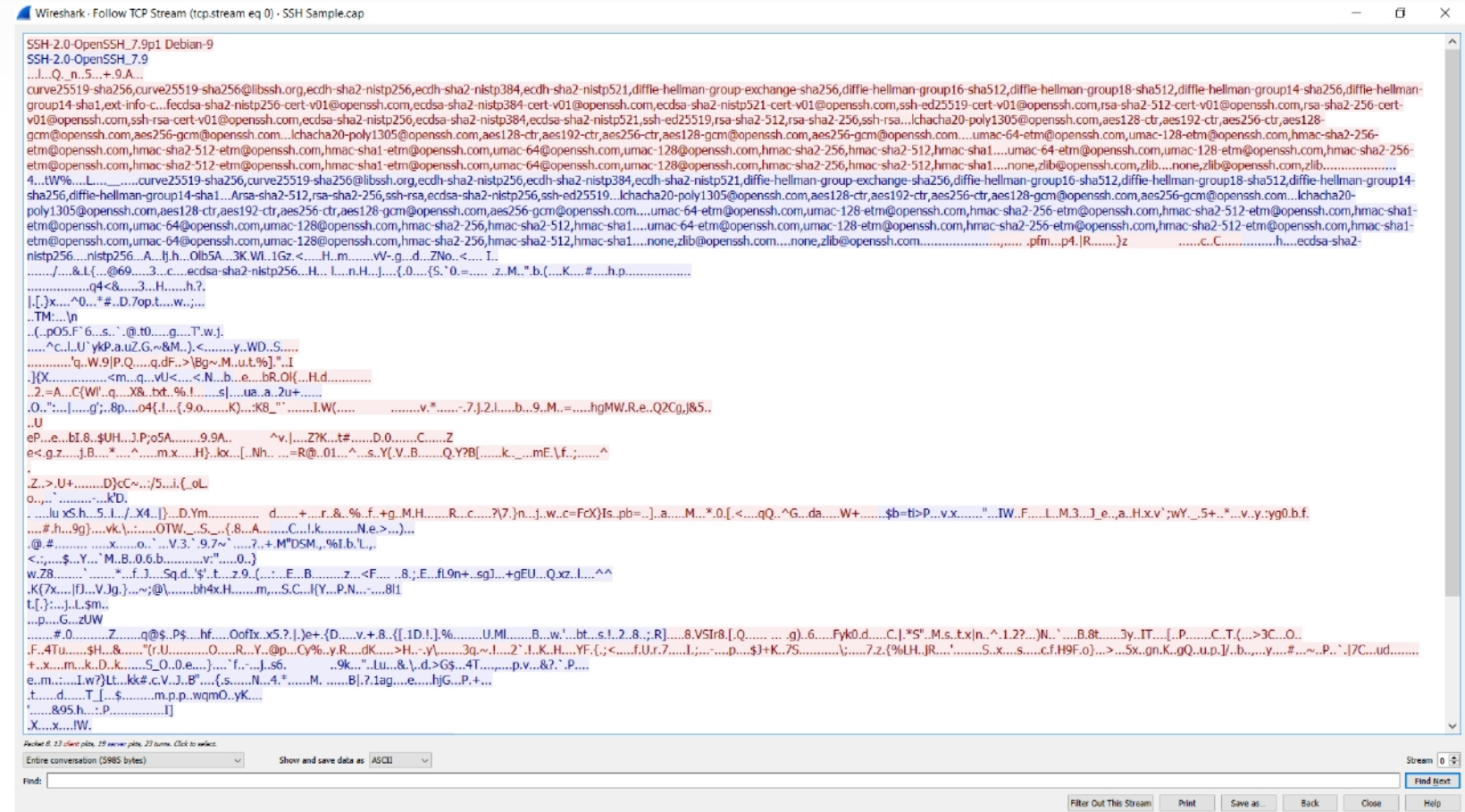
Wireshark



Encrypted Traffic

Encrypted traffic will not be readable, and will appear as non-meaningful characters.

Wireshark



LAB NET-02-L1

Network Examination with Wireshark
30–60 Min.



Mission

The objective of this lab is to learn how to use Wireshark and how computers encapsulate and send data.

Steps

- Become familiar with Wireshark.
- Examine live Network.
- Capture and analyze.

Environment & Tools

- Windows 7 / 8 / 10
- Wireshark Installation file
- Web Browser

Related Files

- Lab document

Network Fundamentals

ARP

ARP

Address Resolution Protocol



If a device wants to communicate with another device on the same LAN, it must encapsulate the MAC address of the destination device. To get the MAC address, it uses a protocol called ARP, that resolves a logical address (IP) to a physical address (MAC), and saves the results in the host ARP cache.

ARP operates in Layers 2 and 3 of the OSI model.

Every Layer 3 device uses ARP.



Switches make forwarding decisions based on the physical address of the frame.

The ARP Process

Scenario: Host A wants to communicate with host B. Host A knows host B's IP address but not its MAC address.

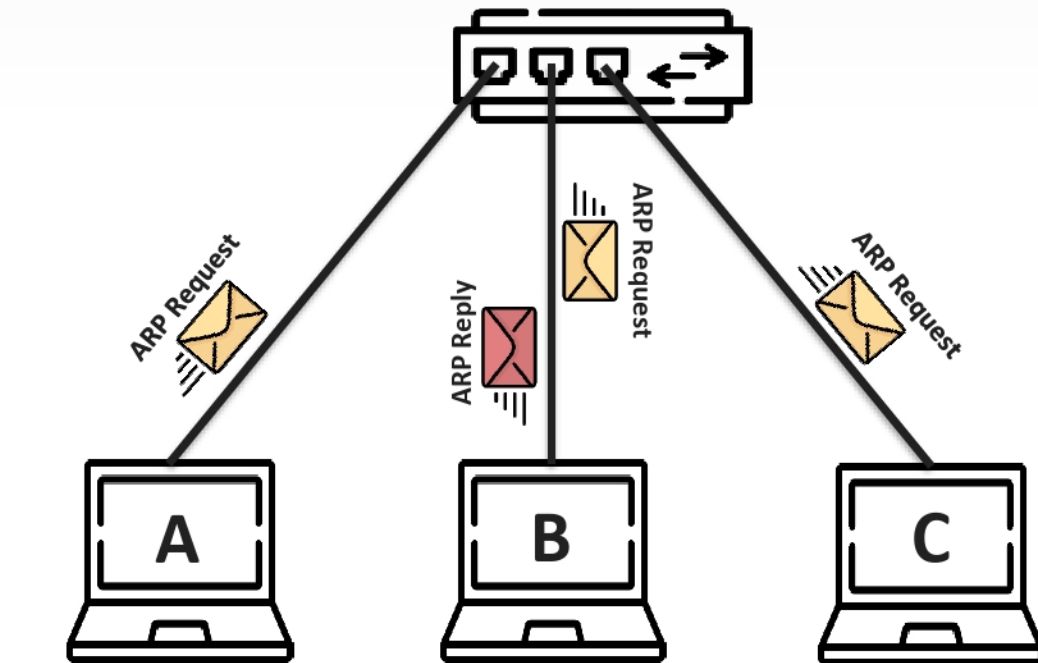
Step 1 - Host A sends an **ARP Request** to all hosts on the network (**Broadcast**), with Host B's IP as the destination MAC, requesting its MAC address.

Step 2 - The switch forwards the ARP request to all ports.

Step 3 - Host B recognizes the IP address in the request, so it performs an ARP reply, in which it lists its own MAC address.

Step 4 - Host A receives the ARP reply and saves host B's MAC address in the ARP cache, together with Host B's corresponding IP address.

ARP Process



ARP Table:

Host	IP	MAC
B	10.0.0.2	B:B:B
C	10.0.0.3	N/A



ARP Table (Cache)

Command

```
arp -a
```



The ARP table or cache is stored in the RAM of the device.

The ARP contains three parameters:

- Destination host IP
- Corresponding MAC of that host
- Entry source MAC learned automatically or manually

arp -a and **arp -g** display the host ARP table.

Show arp displays a Cisco device ARP table.

ARP Table

```
C:\Windows\system32>arp -a
```

```
Interface: 10.71.0.110 --- 0x17
```

Internet Address	Physical Address	Type
10.71.0.100	90-7c-ac-a9-a2-c9	dynamic
10.71.0.104	64-cc-24-96-8b-e0	dynamic
10.71.0.255	ff-ff-ff-ff-ff-ff	static
224.0.0.22	01-00-5e-00-00-16	static
224.0.0.251	01-00-5e-00-00-fb	static
224.0.0.252	01-00-5e-00-00-fc	static
239.255.255.250	01-00-5e-7f-ff-fa	static
255.255.255.255	ff-ff-ff-ff-ff-ff	static



Additional ARP Commands

arp -d deletes the host ARP table.

arp -s adds a static entry to the ARP table that resolves to a specific IP address.
Note: Static ARP helps reduce the risk of ARP spoofing attacks.

arp inet_addr – defines a specific IP address.
arp eth_addr - defines a specific MAC address.

CMD ARP Commands

```
C:\Windows\system32>arp -d
```

```
C:\Windows\system32>arp -s 10.0.0.2 00-aa-bb-11-22-33
```



ARP and Encapsulation

Terminology

Headers & Trailers



Headers and Trailers are control data added by protocols to the beginning and end of each data unit, in each layer.

For example, IP packages the logical source and destination address in an IPv4 header, and adds it to the data unit in the Network Layer.

Encapsulation



Adding Headers and Trailers to data is called encapsulation. Processing them and removing them is called decapsulation.

Depending on the communication type, protocols in each layer add the correct header to the data unit.

PDU



Protocol Data Unit is a piece of information added to or removed from a layer of the OSI model.

PDU consists of protocol-specific control information and user data.

Examples are segments, packets, and frames.

An overhead, high-angle shot of three people sitting around a rustic wooden table. The person on the left is pointing at a laptop screen with a blue pen. The person in the center is typing on a laptop. The person on the right is also working on a laptop. There are two coffee cups, a smartphone, and some papers on the table. The image has a blue tint and a semi-transparent blue overlay on the left side.

QUESTIONS?



THANK YOU
