Capgemini

Module 6 – Networking

Module Learning Objectives

Upon completion of this module, the student should be able to do the following:

- Understand the Open Systems Interconnection (OSI) Model and how information is transferred through the communications stack and back down.
- Understand how information packets are structured and the different protocols and ports that are used to transfer information from one computer or device to another.

Network Basics

Capgemini

Agenda

Open Systems Interconnection (OSI) Model | Routing

Topic Learning Objectives

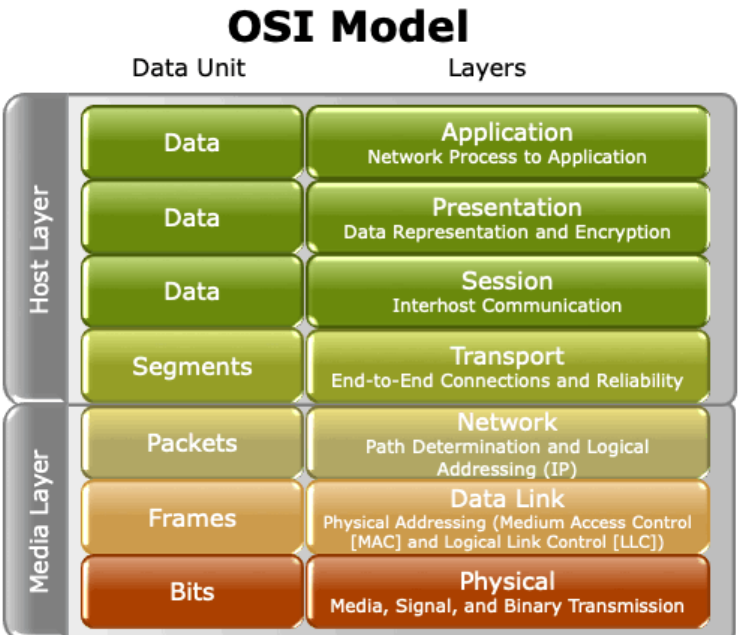
Upon completion of this topic, the student should be able to do the following:

- Describe the OSI Model and Transmission Control Protocol (TCP)/Internet Protocol (IP) Model.
- Describe the OSI Model layers and the devices and protocols associated with each layer.

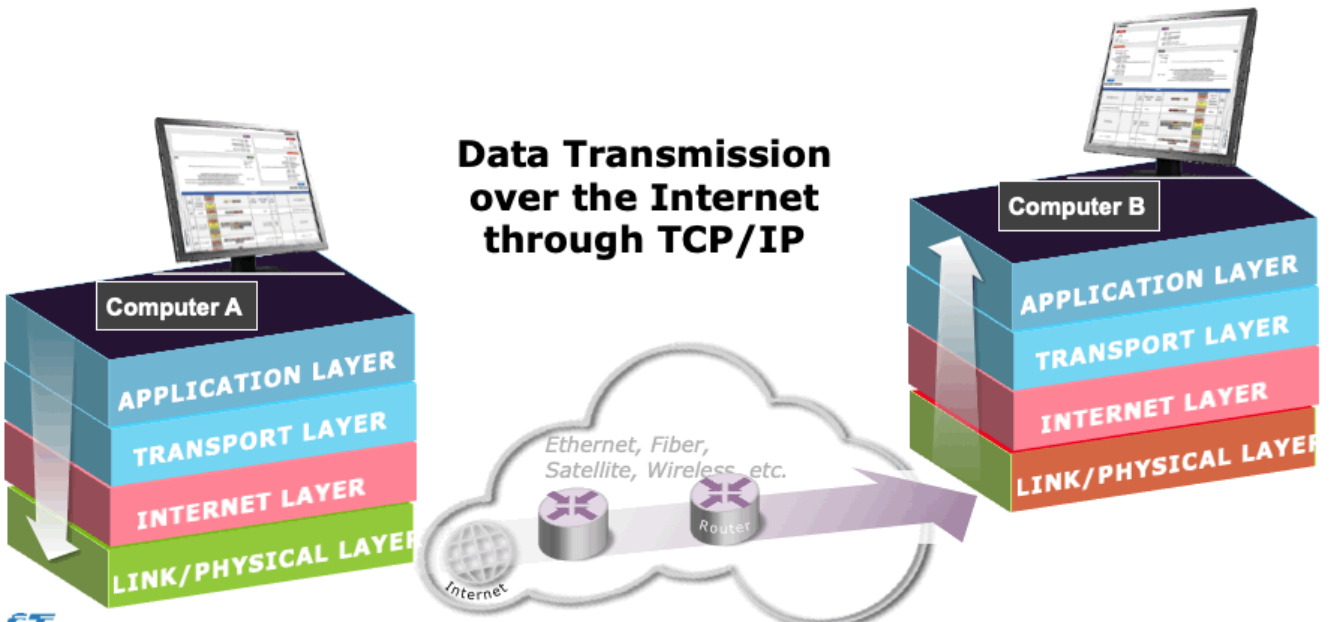
Why is this Important?

- Protocol analysis refers to understanding the protocols and how they work to understand better network operations, including security operations.
- While many of the security tools will interpret much of the protocol data and packets, it is still beneficial to understand the protocols, how they are used, and how they can be maliciously used.

OSI Reference Model



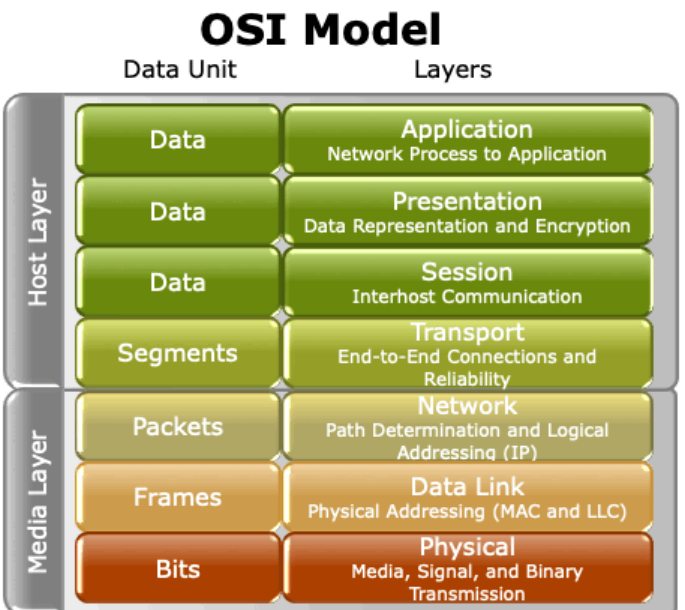
TCP/IP Reference Model



Physical Layer (Layer 1)

OSI Layer

The Physical Layer is where the bits are moved along the wire and consists of the physical transport of information.

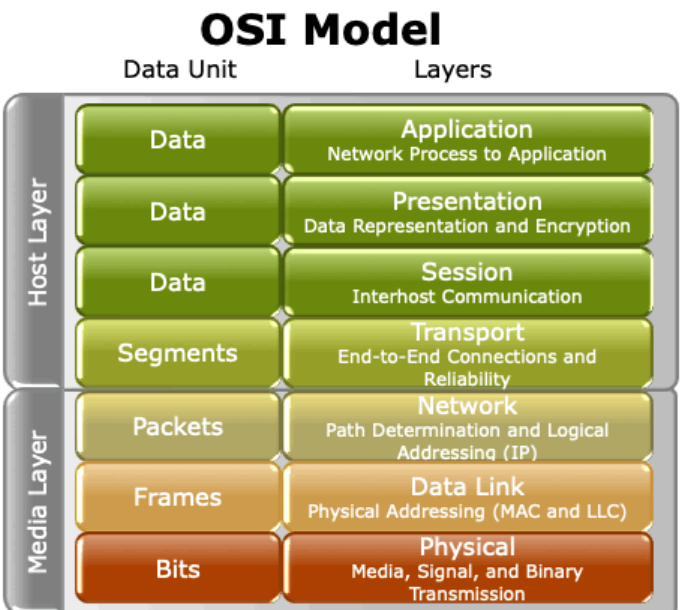


Data Link Layer (Layer 2)

OSI Layer

The Data Link Layer (Layer 2) sublayers are as follows:

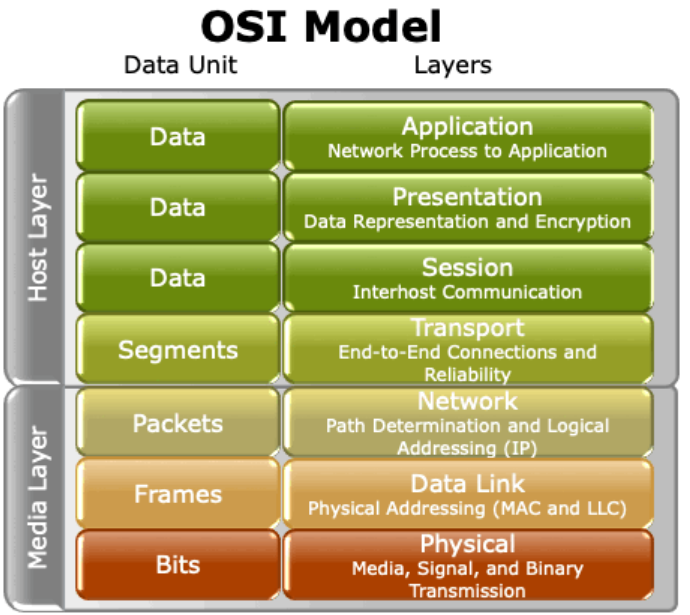
- Medium Access Control (MAC)
- Logical Link Control (LLC)



Data Link Layer (Layer 2) (cont.)

Devices

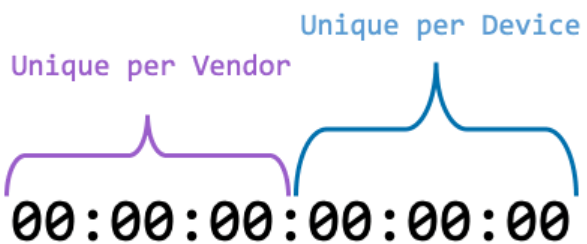
- Hosts
  - Use Network Interface Cards (NICs)
- Hubs
  - Connect multiple hosts together
  - Share the network bandwidth
- Switches
  - Connect multiple hosts together
  - Dedicated network bandwidth
  - Learn MAC addresses seen on ports and build a local MAC address table



Data Link Layer (Layer 2) (cont.)

MAC Addresses

- Important protocols
- 48 bits long
- The first 24 bits are unique for every vendor.
- The second 24 bits are unique for every vendor's device.

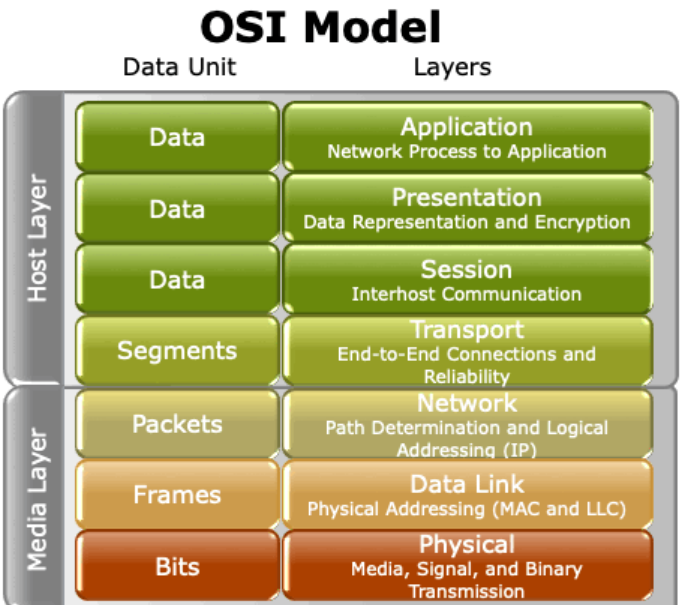



```
Windows IP Configuration

   Host Name . . . . . . . . . . . . : ma6476-LT
   Primary Dns Suffix  . . . . . . . : industrialdefender.com
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : industrialdefender.com
                                       cal

Ethernet adapter Local Area Connection* 13:

   Connection-specific DNS Suffix  . : industrialdefender.com
   Description . . . . . . . . . . . : Juniper Networks Virtual Adapter
   Physical Address. . . . . . . . . : 04-09-85-8F-EB-81
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::604a:7bd0:894c:49b0%24(Preferred)
   IPv4 Address. . . . . . . . . . . : 172.16.30.198(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 0.0.0.0
   DHCPv6 IAID . . . . . . . . . . . : 1006765445
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-23-48-4C-1E-20-47-47-D6-09-31
   DNS Servers . . . . . . . . . . . : 172.16.35.10
                                       172.16.35.11
   NetBIOS over Tcpip. . . . . . . . : Enabled
```


Data Link Layer (Layer 2) (cont.)

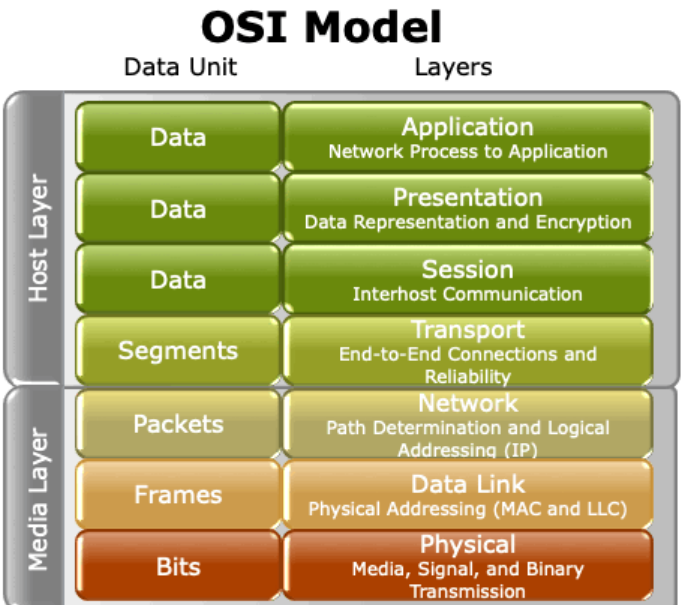
Address Resolution Protocol (ARP)

- Layer 2 protocol that lets devices query the network for the MAC address of a specific IP address
- Host devices use ARP to build a local ARP table, a cache of IP to MAC mappings.



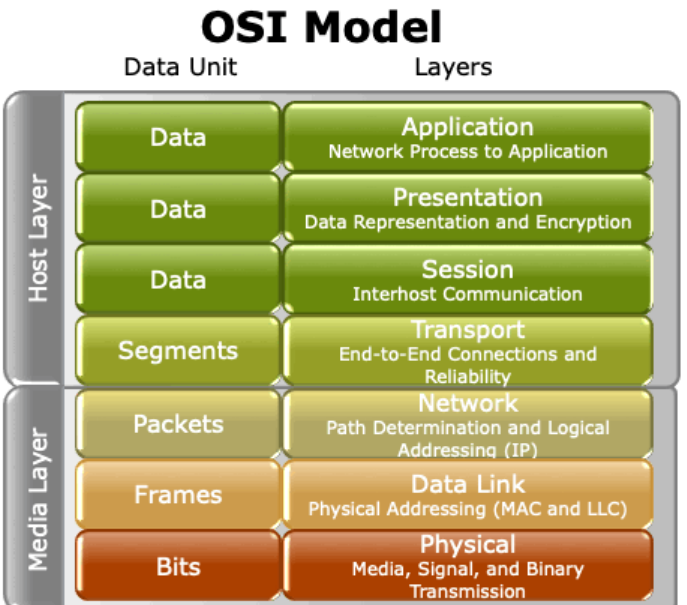
Network Layer (Layer 3)

- IP is at the Network Layer
- Internet Protocol Version 4 (IPv4) – 32-bit address
- Internet Protocol Version 6 (IPv6) – 128-bit address



Network Layer (Layer 3) (cont.)

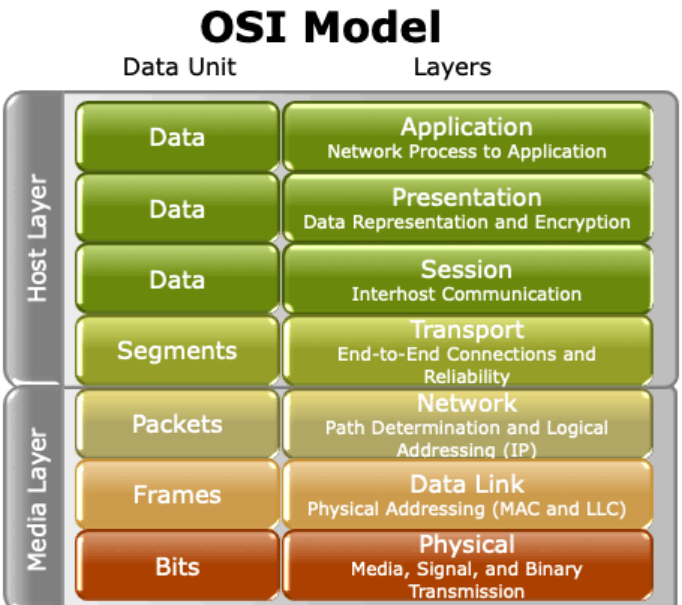
- Devices
  - Routers
  - Connect multiple subnets together



Transport Layer (Layer 4)

Responsible for end-to-end communication over a network

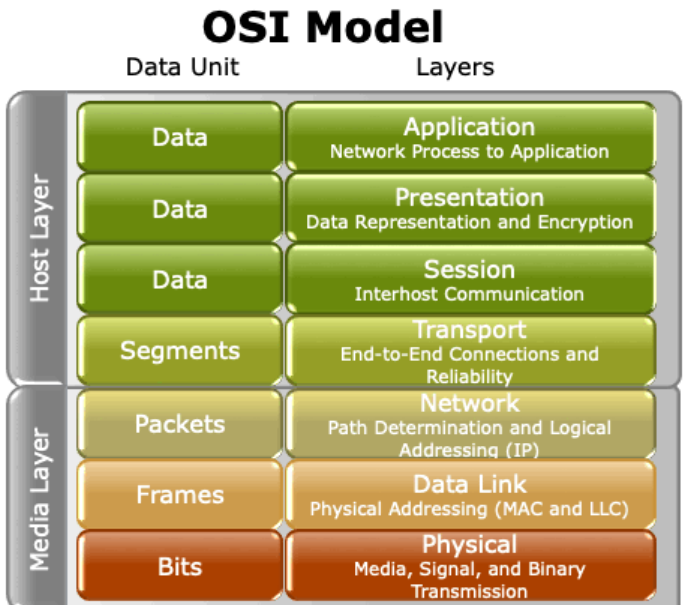
- Provides logical communication between application processes on different hosts
- Provides a point-to-point connection rather than one hop to another hop (or router)
- Connection-Oriented Service
- TCP
- Connectionless Service
- User Diagram Protocol (UDP)



Session Layer (Layer 5)

Responsible for opening and closing sessions between end user application processes

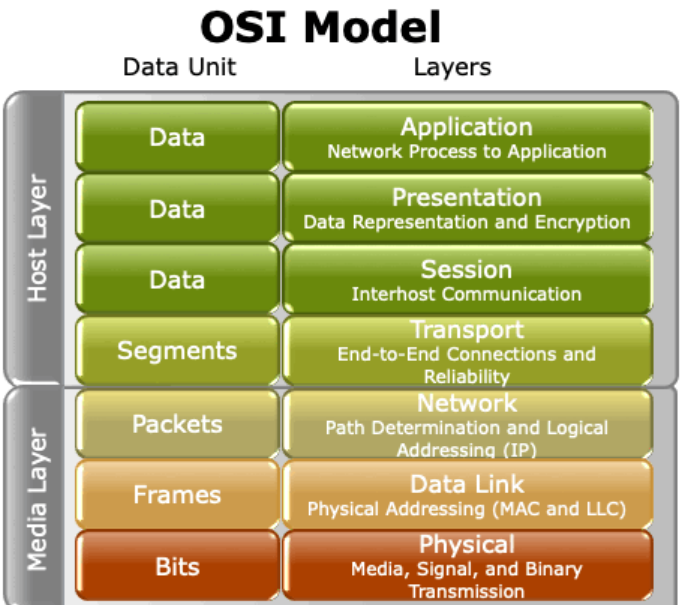
- Controls single or multiple application connections
- Sessions are commonly implemented on web browsers.
- Creates procedures for check pointing, adjournment, restart, and termination
- Responsible for synchronizing information from different sources
- Supports full-duplex and half-duplex operations



Presentation Layer (Layer 6)

Responsible for the delivery and formatting of information

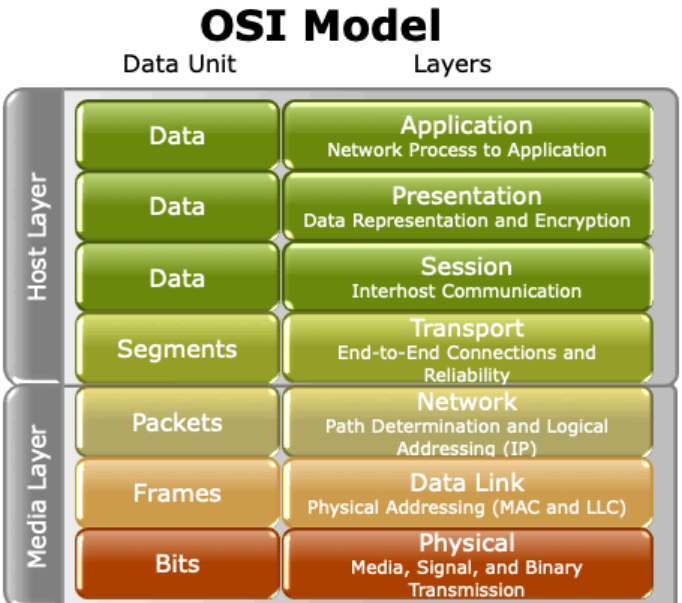
- Handles data representation differences
- Data conversions
- Encryption



Application Layer (Layer 7)

Provides services to the application programs

- Ensures the availability of the receiving device to receive application data
- Enables authentication, if necessary
- Ensures error recovery, data integrity, and privacy
- Presents the data to the user application



IP Routing

Routers forward IP traffic between subnets.

- They use a routing table to determine the next hop or port to send the data out.
- Two types of IP routing are as follows:
  - Static: Entries in the routing table and pre-defined
  - Dynamic: Dynamically reconfigure the routing table to find the "best" route

Protocol	Type	Location
Routing Information Protocol (RIP) v1 and v2	Distance Vector	Local Area Networks (LANs)
Open Shortest Path First (OSPF)	Link State	LANs, Small Wide Area Networks (WANs)
Interior Gateway Routing Protocol (IGRP)	Distance Vector	LANs; Cisco Proprietary
Enhanced Interior Gateway Routing Protocol (EIGRP)	Hybrid	LANs; Cisco Proprietary
Border Gateway Protocol (BGP)	Distance Vector	WANs

Common Network and Security Protocols

There are numerous protocols become familiar with when learning more about protocols and their common vulnerabilities (included is a brief list of some of the many protocols).

Application Layer Protocols

DHCP: Dynamic Host Configuration Protocol
DNS: Domain Name System (Service) Protocol
FTP: File Transfer Protocol
HTTP: Hypertext Transfer Protocol
IMAP and IMAP4: Internet Message Access Protocol (version 4)
IRCP: Internet Relay Chat Protocol
LDAP: Lightweight Directory Access Protocol (version 3)
MIME (S-MIME): Multipurpose Internet Mail Extensions and Secure MIME
NAT: Network Address Translation
POP and POP3: Post Office Protocol (version 3)
SMTP: Simple Mail Transfer Protocol
SNMP: Simple Network Management Protocol
TELNET: Terminal Emulation Protocol of TCP/IP
TFTP: Trivial File Transfer Protocol
URL: Uniform Resource Locator

Presentation Layer Protocols

LPP: Lightweight Presentation Protocol

Session Layer Protocols

RPC: Remote Procedure Call Protocol

Transport Layer Protocols

TCP: Transmission Control Protocol
UDP: User Datagram Protocol

Data Link Layer Protocols

ARP: Address Resolution Protocol
RARP: Reverse Address Resolution Protocol
Kerberos: Network Authentication Protocol
RADIUS: Remote Authentication Dial-in User Service

Common Network and Security Protocols (cont.)

Network Layer Protocols

- BGP (BGP-4): Border Gateway Protocol
- EGP: Exterior Gateway Protocol
- IP: Internet Protocol (IPv4) and (IPv6)
- ICMP and ICMPv6: Internet Message Control Protocol
- IRDP: ICMP Router Discovery Protocol
- Mobile IP: IP Mobility Support Protocol for IPv4/IPv6
- OSPF: Open Shortest Path First Protocol
- RIP: Routing Information Protocol
- RSVP: Resource Reservation Protocol
- BGMP: Border Gateway Multicast Protocol
- DVMRP: Distance Vector Multicast Routing Protocol
- IGMP: Internet Group Management Protocol
- MARS: Multicast Address Resolution Server
- MBGP: Multiprotocol BGP
- MSDP: Multicast Source Discovery Protocol
- MPLS: Multiprotocol Label Switching

Other Protocols

- FDDI: Fiber Distributed Data Interface
- Token Ring: Institute of Electrical and Electronics Engineers (IEEE) 802.5 LAN Protocol
- LLC: Logic Link Control (IEEE 802.2)
- SNAP: SubNetwork Access Protocol
- STP: Spanning Tree Protocol (IEEE 802.1D)
- SCSI: Small Computer System Interface

Local Area Network and LAN Protocols
Virtual LAN Protocols
Wireless LAN Protocols
Metropolitan Area Network and MAN Protocol

Wide Area Network Protocols

- ATM: Asynchronous Transfer Mode
- SONET/SDH: Synchronous Optical Network and Synchronous Digital Hierarchy

Broadband Access Protocols

- BISDN: Broadband Integrated Services Digital Network (Broadband ISDN)
- ISDN: Integrated Services Digital Network
- Frame Relay: WAN Protocol for Internetworking
- HDLC: High-Level Data Link Control

Common Network and Security Protocols (cont.)

Signaling Protocols

H.323: Voice over Internet Protocol (VOIP)
H.235: Security and Encryption for H-Series

Point-to-Point Protocols

PPP: Point-to-Point Protocol
EAP: PPP Extensible Authentication Protocol
CHAP: Challenge Handshake Authentication Protocol
LCP: PPP Link Control Protocol
MPPP: Multilink Point to Point Protocol (MultiPPP)
PAP: Password Authentication Protocol

Tunneling Protocols

L2F: Layer 2 Forwarding
L2TP: Layer 2 Tunneling Protocol
PPTP: Point-to-Point Tunneling Protocol

Secured Routing Protocols

GRE: Generic Routing Encapsulation
IPSec: Security Architecture for IP
IPSec AH: IPsec Authentication Header
IPsec ESP: IPsec Encapsulating Security Payload
IPsec IKE: Internet Key Exchange
IPsec ISAKMP: Internet Security Association and Key Management Protocol
TLS: Transport Layer Security
Many Other Security Protocols

Cisco Protocols

CDP: Cisco Discovery Protocol
CGMP: Cisco Group Management Protocol
DTP: Cisco Dynamic Trunking Protocol
EIGRP: Enhanced Interior Gateway Routing Protocol
HSRP: Hot Standby Router Protocol
IGRP: Interior Gateway Routing Protocol
ISL and DISL: Cisco Inter-Switch Link Protocol and Dynamic ISL Protocol
RGMP: Cisco Router Port Group Management Protocol
TACACS (and TACACS+): Terminal Access Controller Access Control System
VTP: Cisco VLAN Trunking Protocol
XOT: X.25 over TCP by Cisco

Network Models

Takeaways

- It is important to understand at what layers in the OSI Model the network security controls operate.
- Most Advanced Persistent Threats (APTs) reside at the Application Layer

Lower Layers

- Faster, easier to collect and analyze data
- Useful for statistical and endpoint analysis

Higher Layers

- Larger data sets, more complicated to analyze (e.g., full packet capture, deep packet inspection)
- Critical for detection and mitigation of advanced threats

Questions?

Analyzing Network Packets

Capgemini

Agenda

Packet Analysis | Wireshark |
Command Line Interface (CLI) Network Analysis Tools

Topic Learning Objectives

Upon completion of this topic, the student should be able to do the following:

- Understand the Wireshark interface and how it is used.
- Understand different command-line tools that can be used to conduct packet analysis.
- Understand how packets are analyzed and how Wireshark can be used to conduct packet analysis.
- Understand how to analyze and replay packets using different command-line tools.
- Understand how files can be extracted from Packet Capture (PCAP) files and how this can assist forensics efforts.

Importance of Packet Analysis

Cybersecurity Breach

- When and how did the malicious activity begin?
- Are there indicators that the malicious activity is ongoing?
- What was the impact of the breach (systems affected, data taken, access, etc.)?
- Was sensitive or confidential information taken?
- What are the regulatory, legal, and privacy concerns associated with the breach?

Packet Analysis Overview

Why conduct packet analysis?

- Analyze packets that cross the network
- Data in the packets
- Identify changes to the packets and associated content

- Network Management
- Fault Management
- Network Security
- Malicious Actors

Overview of Packet Analysis Tools

There are many packet analysis tools available

Common capabilities include the following:

- Network troubleshooting
- Malware detection and identification
- Traffic baselines, metrics, and pattern analysis
- Identify protocol activity and unused protocols
- Monitor network traffic (normal and malicious)

Wireshark

Graphical User Interface (GUI) tool for packet capture and analysis

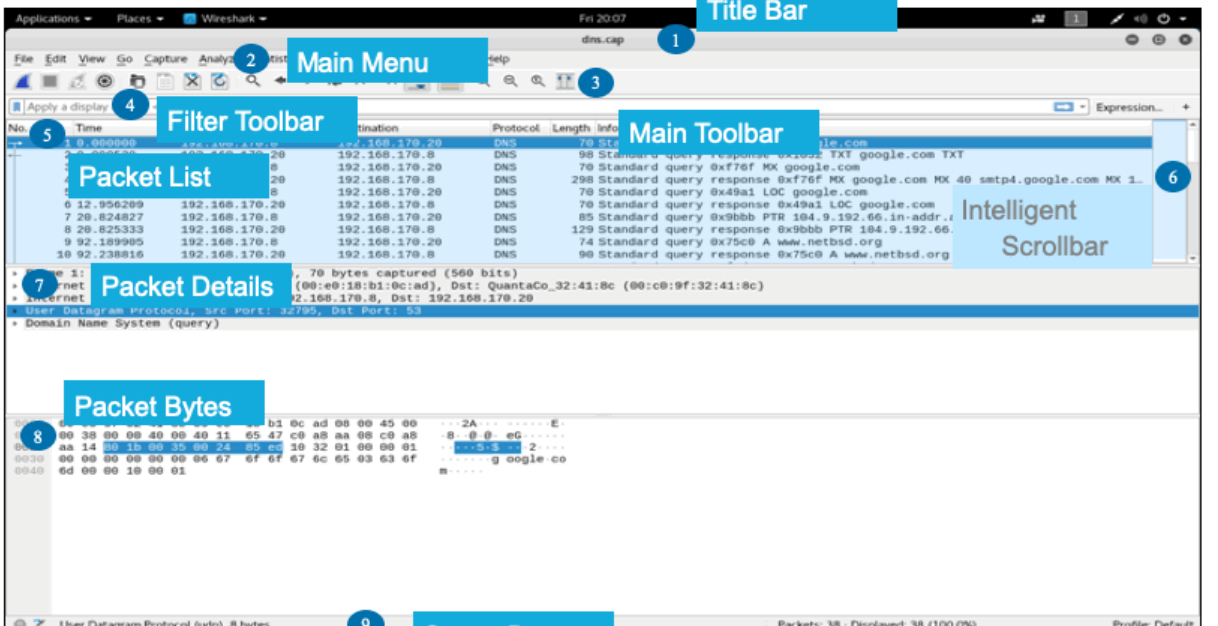
- Open source
- Available for *nix and Windows
- Protocol analysis capabilities – much more than just a PCAP and visualization tool
- Used extensively in network management activities
- Wireshark can import many other file formats, but the most common file format is "tcpdump" (PCAP).

Wireshark (cont.)

Wireshark Capabilities

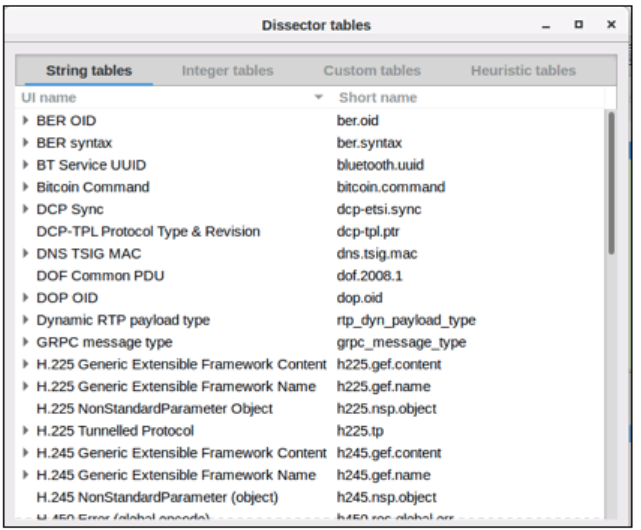
- Deep inspection of hundreds of protocols, with more being added all the time
- Live capture and offline analysis
- Standard three-pane packet browser
- Multi-platform: Runs on Windows, Linux, OS X, FreeBSD, NetBSD, and many others
- Captured network data can be browsed via a GUI or via the TTY-mode TShark utility
- The most powerful display filters in the industry
- Rich VoIP analysis
- Read/Write many different capture file formats
- Capture files compressed with gzip can be decompressed on the fly
- Live data can be read from Ethernet
- Decryption support for many protocols
- Coloring rules can be applied to the packet list for quick, intuitive analysis
- Output can be exported to Extensible Markup Language (XML), PostScript, Comma-Separated Value (CSV), or plaintext

Wireshark GUI



Wireshark

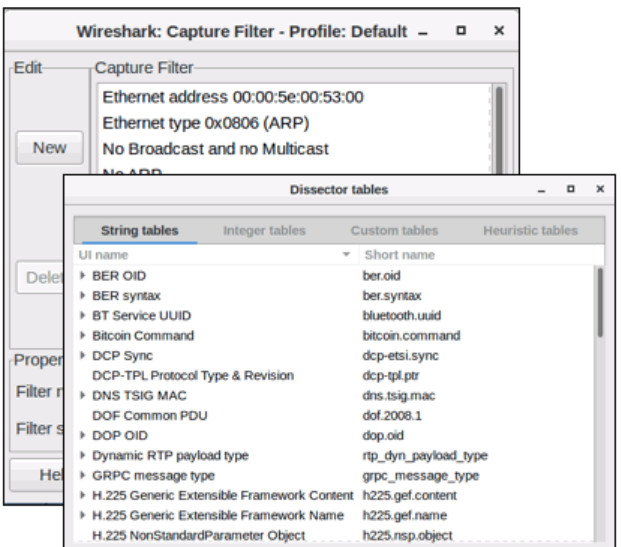
- Protocol Dissectors
  - Wireshark understands protocol formats via hundreds of protocol dissectors.
  - Dissectors enable Wireshark to parse bits in each packet.
  - Wireshark interprets the dissector for each packet based on the following:
    - Static assignments (defined by the user)
    - Port number
  - Fields can then be used to filter results.



Wireshark Filters

Wireshark has two types of filters (Capture and Display)

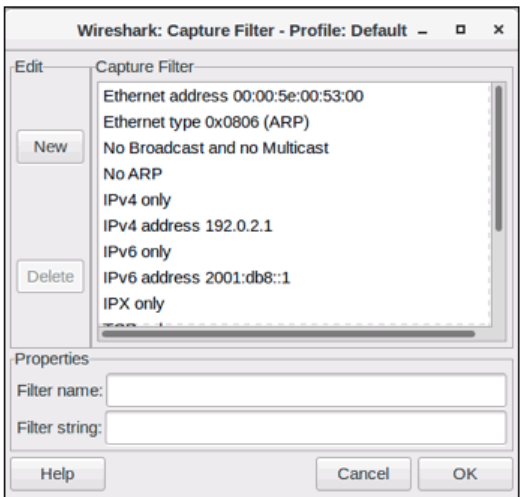
- Capture filters: Used in the Wireshark Capture Options screen. Examples are as follows:
  - Capture only traffic to or from IP address 122.118.5.4:
    - host 122.118.5.4
  - Capture traffic to or from a range of IP addresses:
    - net 210.118.0.0/24
  - Capture traffic from a range of IP addresses:
    - src net 210.118.0.0/24
  - Capture traffic to a range of IP addresses:
    - dst net 210.118.0.0/24
  - Capture traffic within a range of ports:
    - tcp portrange 1401-1499



Wireshark Filters (cont.)

Display filters: From the main display screen

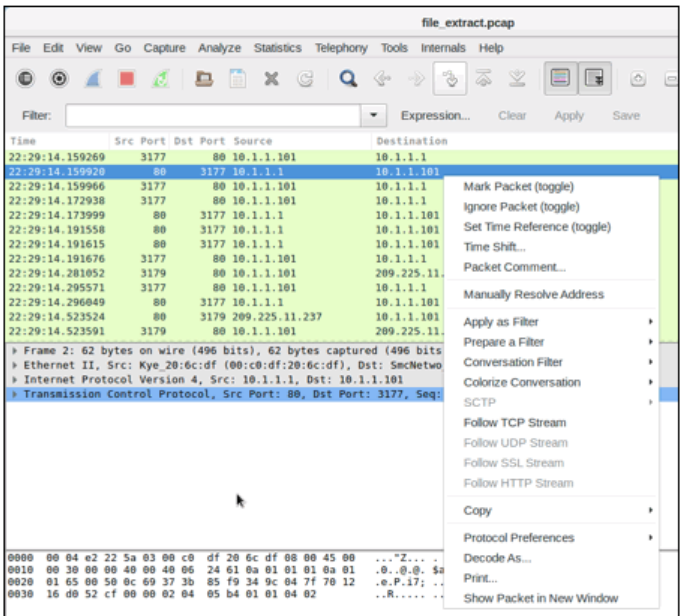
- Examples are as follows:
  - Show only SMTP (port 25) and ICMP traffic:
    - tcp.port eq 25 or icmp
  - Show only traffic in the 192.168.0.0/16 subnet:
    - ip.src==192.168.0.0/16 and ip.dst==192.168.0.0/16
  - Match HTTP requests for PHP pages:
    - http.request.uri matches "php$"



Wireshark Protocol Dissector Example

Protocol Dissectors

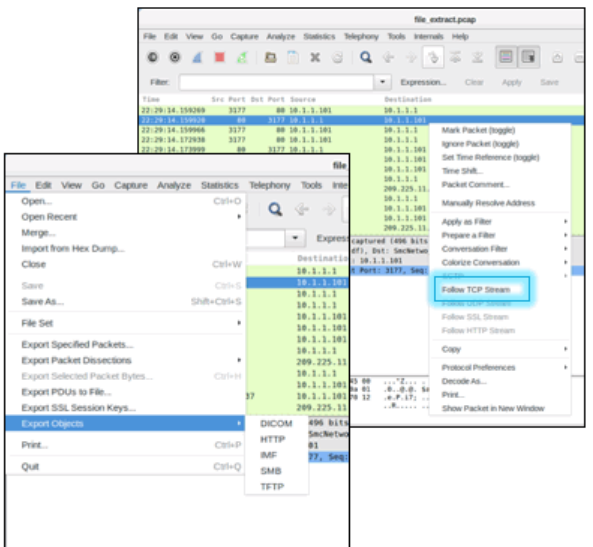
- Frame
- Ethernet
- IPv4
- TCP
- HTTP
- Right-click on the Protocol Filter to display the pop-up.
- Choose “Filter Field Reference” to see the fields defined for the selected protocol.



Wireshark Streams

Rebuilding Data Streams and Extracting Files

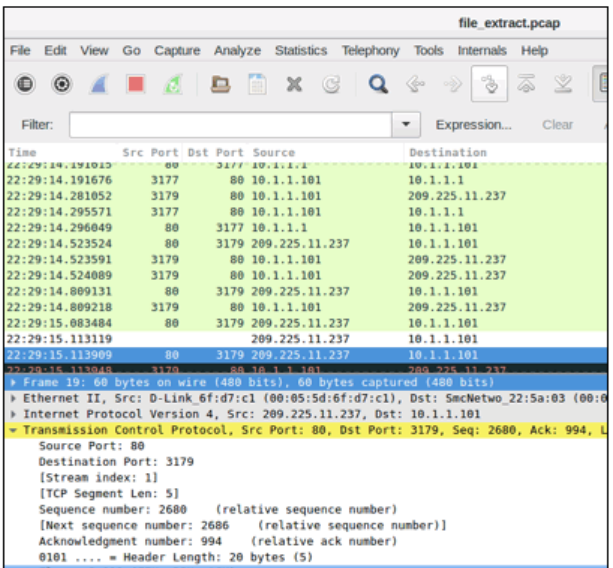
- Wireshark will reconstruct the raw data and display it as the application sees it.
- Do this with the "Follow TCP Stream" or by using the Export Object Menu.
- Supports different protocols
- Can reconstruct and extract files transferred via defined protocols



Wireshark Experts

Rebuilding Data Streams and Extracting Files

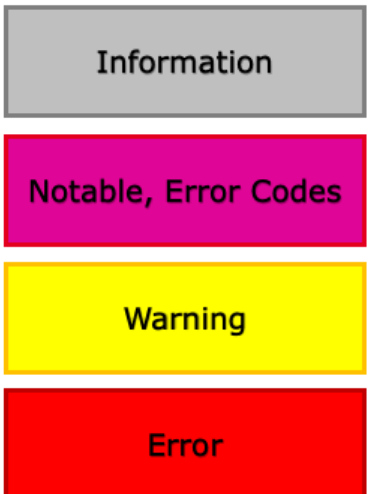
- Built-in diagnostics
- Identify errors, anomalous activity, and other miscellaneous activities
- Fidelity varies by protocol



Wireshark Experts (cont.)

Wireshark Packet Analysis

- Gray: Information (e.g., TCP SYN flag set)
- Cyan: Notable; error codes (e.g., HTTP 404)
- Yellow: Warning (e.g., out of order segment)
- Red: Error (e.g., malformed packet)



tshark

Tshark CLI Version of Wireshark

- Compatible with Windows and Linux distributions
- Supports same options and provides many of the same capabilities as Wireshark
- Tshark is scriptable

Tshark Options

Useful Options:

-f <FILTER>	Specify a capture filter (**Note:** different from a display filter)
-i <INTERFACE>	Specify interface/pipe to use for live captures
-t <FORMAT>	Specify format of timestamp
	("ad" = absolute with date, "r" = relative, etc.)
-n	Disable network object name resolution
	(DNS, port names, etc.)
-r <INFILE>	Read packet data from <INFILE> instead of interface
-R <FILTER>	Specify a display filter
	(similar to the Filter: field in Wireshark)
	Packets that do not match are discarded
-S	Decode and display packets
-w <OUTFILE>	Write capture data to <OUTFILE>

Tshark Example

- Capture packets, and display only packets from 10.12.200.0/24.
  - # tshark –i eth2 -t ad –n –S –R "ip.addr==10.12.200.0/24"

- Read in file "web.pcap," display only HTTP traffic that contains the string ".pdf," and save the results to a new file.
  - # tshark –r web.pcap –S –R "http and frame contains '.pdf'" –w pdfs.pcap

Lab005: Introduction to Wireshark

Lab006: Customizing Wireshark

tcpdump

Popular PCAP and Analysis CLI Tool

- Outputs to stdout or to file in tcpdump format: -w <filename>
- Reads network traffic: From interface (“-i <interface>”) or from a file (“-r <filename>”)

Common Flags:

i <INTERFACE>	Capture from <INTERFACE>
-r <FILE>	Read in packets from <FILE>
-w <FILE>	Write captured packets to <FILE>
-n	Do not resolve hostnames (very useful when need to filter by IP address)
-nn	Do not resolve hostname or port names
-X	Show packet contents in hex and American Standard Code for Information Interchange (ASCII)
-XX	Same as –X, but also show the Ethernet header
-v[v,vv]	Show increasing amount of detail
-E <key>	Decrypts IPsec traffic (if have the key)
-s	Set the amount of data returned (default is 96 bytes)

tcpdump Example

Output DNS traffic (port 53) from host 210.118.10.101:

- # tcpdump -r 2.pcap -nnvvXS udp and port 53 and src 210.118.10.101

Output only the IP and TCP headers for traffic from 213.54.69.0/24:

- # tcpdump -r 2.pcap -nnv net 213.54.69.0/24 and tcp

Output and sort the unique IP addresses:

- # tcpdump -r 2.pcap -nnv net 213.54.69.0/24 and tcp | grep -oE '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}' |sort -u

Searching and Merging

Command-Line PCAP Searching

- ngrep stands for network grep.
- ngrep provides the capability to search text-based data and non-text data.
- Syntax:
  - ngrep [options] –I <infile> "search pattern regex" "filter"

Examples:

- Search a PCAP for instances of username "jdoe":
- #ngrep –I eth0.pcap "jdoe"
- Search a PCAP for hex characters that represent a executable file header; output matching packets to a new PCAP called "evidence.pcap":
- #ngrep –X –I eth0.pcap –O evidence.pcap "4D5A"

Searching and Merging (cont.)

Search for outgoing web requests from 12.13.14.15:

- #ngrep –I eth0.pcap "^(GET|POST)" "src host 12.13.14.15 and tcp and dst port 80"

Search for cleartext POP3 logins:

- #ngrep –I eth0.pcap "USER" "tcp and port 110"

Searching and Merging (cont.)

Combining PCAPs – two methods (Wireshark or mergecap):

- Wireshark
  - File -> Merge
  - Choose PCAP to merge

- Mergecap – command-line tool, included in Wireshark suite on both *nix and Windows versions
  - Syntax:
    - –mergecap [options] –w <outfile> |- infile1 infile2 infile3 …
  - Example:
    - –#mergecap -w merge.pcap eth0.pcap eth1.pcap eth2.pcap

Editing a PCAP

editcap

- Command-line tool, included in Wireshark suite
- Syntax:
  - editcap [options] infile outfile
- Example – save just the headers for the first 1,000 packets to a new file:
  - #editcap –s 64 -r full_packets.pcap headers.pcap 1-1000

Replaying Packets

tcpreplay can be used to replay packets

- tcpprep: PCAP pre-processor to split traffic into two sides (client, server)
- tcprewrite: Re-map ports, IPs, MAC addresses, and others as needed
- tcpreplay: Put the packets on the wire; change the timing

Replaying Packets (cont.)

Example: Replay client/server traffic through an Intrusion Prevention System (IPS) or other inline device:

- Step 1: Use tcpprep to split the traffic based on source/destination port:
  - # tcpprep --port --cachefile=temp.cache --pcap=old.pcap
- Step 2: Use tcprewrite to change the IP addresses to be on the local network:
  - # tcprewrite --endpoints=192.168.0.10:172.16.0.25 --cachefile=temp.cache --infile=old.pcap --outfile=new.pcap
- Step 3: Send the traffic (use eth0 as the client interface and eth1 as the server interface):
  - # tcpreplay --inf1=eth0 --inf2=eth1 --cachefile=temp.cache new.pcap

Extracting TCP Conversations

tcpflow is a command-line tool to parse, reassemble, and extract payloads of any TCP stream it finds in a libpcap PCAP.

- Example: Use tcpflow to extract TCP flows from IP address 192.168.1.124: # tcpflow –v –r capturefile.pcap 'host 192.168.1.124'

tcpxtract is a libpcap-based tool designed to extract and reconstruct payload data based on file signatures. It contains a configuration file with beginning sequences of known file formats.

- Example: Use tcpxtract to extract all recognizable files from a PCAP to a specific directory: # tcpxtract –f capturefile.pcap –o output_directory/

Snort

Three Modes

1. Network Intrusion Detection System (NIDS) mode (performs network traffic detection and analysis)
2. Sniffer mode (reads network packets and displays them)
3. Packet Logger mode (logs packets to a disk)

Snort (cont.)

Very powerful and customizable

- rule – signature + action
- GUI and CLI
- Real-time detection capability and can replay attacks and test new indicators

Useful Flags:

c <conf file>	Specify which configuration file to use
-r <FILE>	Read in a single PCAP file
-pcap-dir=<dir>	Read in PCAPs recursively from <dir>
-A <mode>	Alert mode (via stdout); common options: full, fast, none
-s	Send alerts to syslog
-T	Test mode; useful for testing the syntax of rules

Snort Rules

Use the Snort configuration file (typically "snort.conf") to enable rules

- Custom configurations can be passed to Snort via the "-c" parameter.
- Examples (from "/etc/snort/snort.conf"):
  - include $RULE_PATH/local.rules
  - include $RULE_PATH/bad-traffic.rules
  - include $RULE_PATH/exploit.rules
  - include $RULE_PATH/community-exploit.rules
  - include $RULE_PATH/scan.rules

Snort Rules (cont.)

Rules are defined in files

- Usually saved in /etc/snort/rules with extension.rules

Example (from "\etc\snort\rules\ftp.rules"):

- alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP CWD ~root attempt"; flow:to_server,established; content:"CWD"; nocase; content:"~root"; distance:1; nocase; pcre:"/^CWD\s+~root/smi"; reference:arachnids,318; reference:cve,1999-0082; classtype:bad-unknown; sid:336; rev:10;)
- alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP CWD ..."; flow:to_server,established; content:"CWD"; nocase; content:"..."; distance:0; pcre:"/^CWD\s[^\n]*?\.\.\./smi"; reference:bugtraq,9237; classtype:bad-unknown; sid:1229; rev:7;)
- alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP CWD ~ attempt"; flow:to_server,established; content:"CWD"; nocase; pcre:"/^CWD\s+~/smi"; reference:bugtraq,2601; reference:bugtraq,9215; reference:cve,2001-0421; classtype:denial-of-service; sid:1672; rev:11;)

LAB007: Carving Files with Wireshark

Questions?

Advanced Networking Concepts

Capgemini

Agenda

TCP/IP and Protocols | Network Devices

Topic Learning Objectives

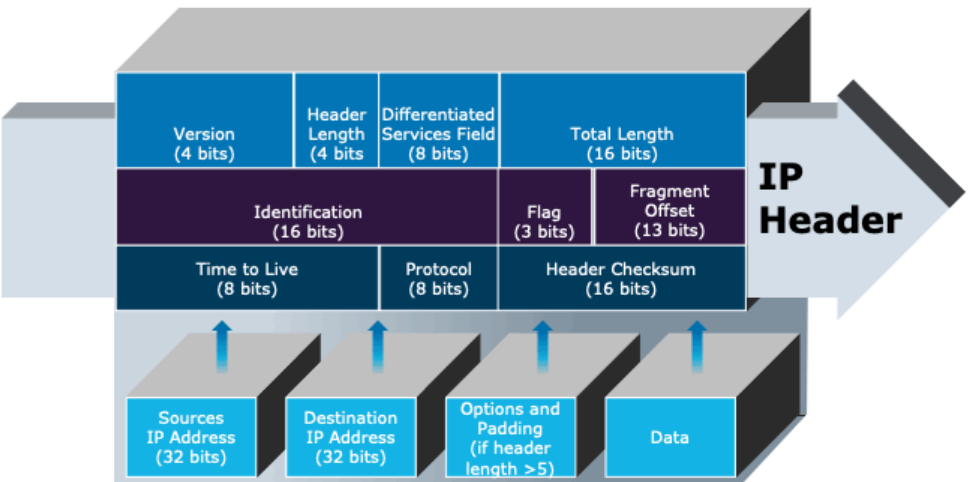
Upon completion of this topic, the student should be able to do the following:

- Describe key TCPs/IPs and their related vulnerabilities.
- Describe network devices, how they protect the network, and their related vulnerabilities.
- Describe network protocols and technologies, how they protect the network, and their related vulnerabilities.

Internet Protocol (IP) – The Basics

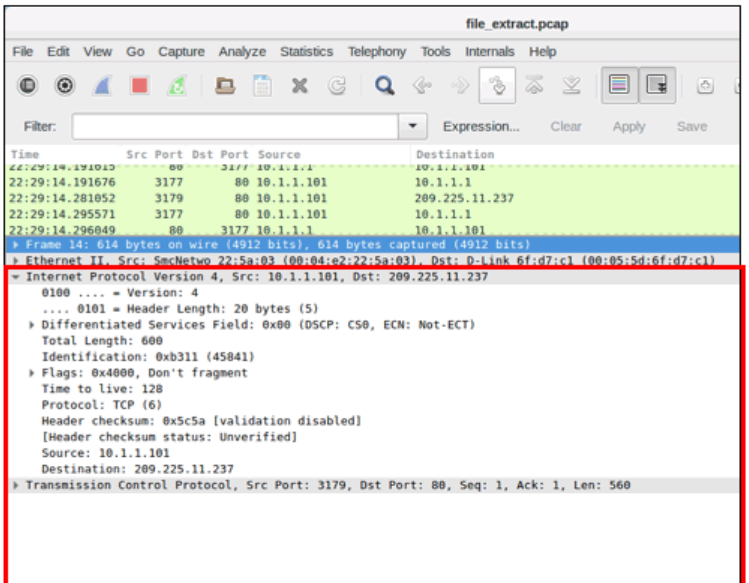
IP contains the fields necessary to route traffic from one network to another. IP is a connectionless protocol.

IPv4 Header:



IPv4 Header – Example

IPv4 Header as seen in Wireshark:

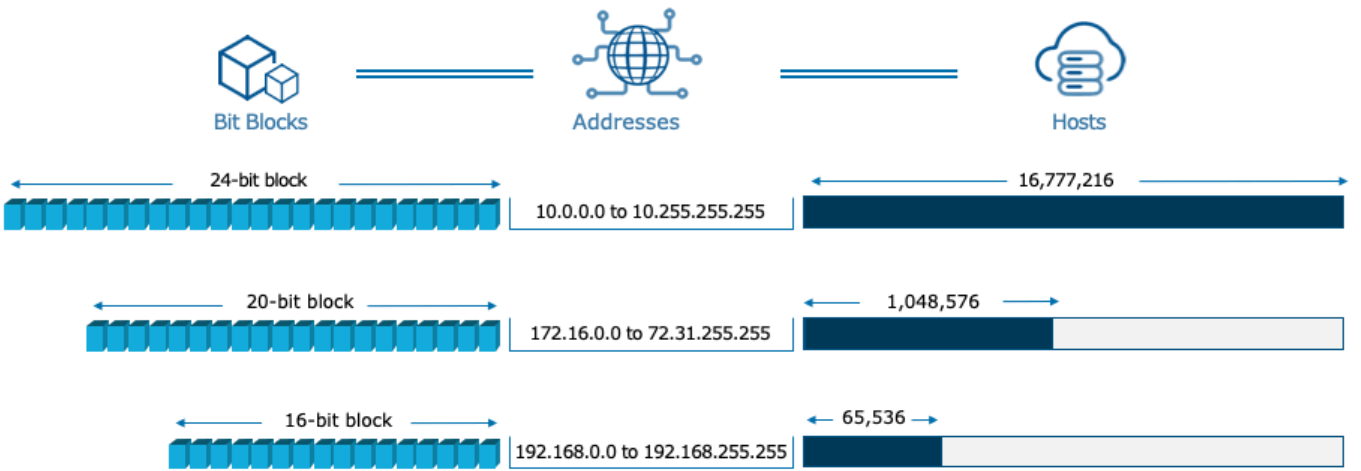


IPv4 – Key Concepts

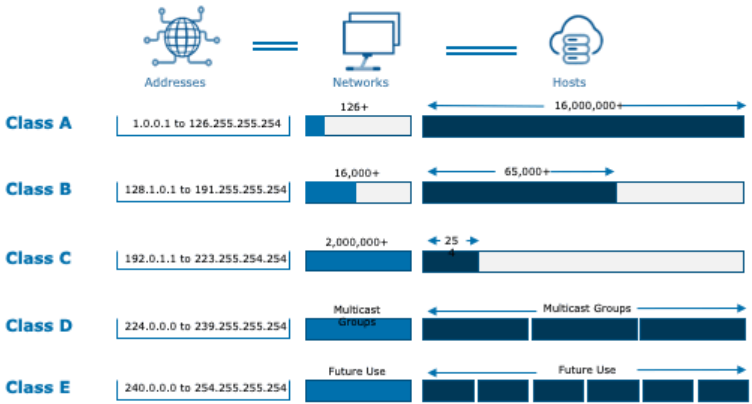
Subnets

- Subnet Masks (such as IP address 198.170.10.120, Class C Subnet Mask 255.255.255.0)
- Classless Inter-Domain Routing (CIDR) notation (such as 198.170.10.120/24)
- Additionally, each network can be further subnetted, such as the 198.170.10.X network could be subnetted into four subnets, with a subnet mask of 255.255.255.192.

RFC1918 Name



IP Address Ranges

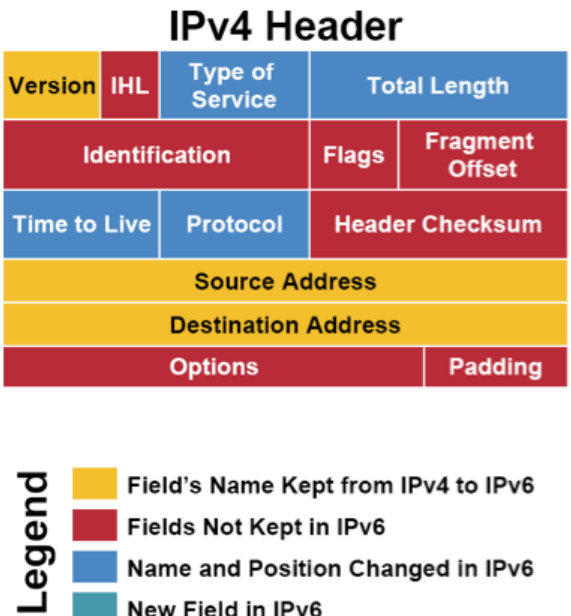


Class	Address Range	# Networks	# Hosts
A	1.0.0.1 to 126.255.255.254	126	16,000,000+
B	128.1.0.1 to 191.255.255.254	16,000+	65,000+
C	192.0.1.1 to 223.255.254.254	2,000,000+	254
D	224.0.0.0 to 239.255.255.254	Multicast Groups	Multicast Groups
E	240.0.0.0 to 254.255.255.254	Future Use	Future Use

Network Layer (Layer 3)

IPv4

- 4 sets of decimal numbers from 0 to 255
- This is referred to as "dotted-decimal" notation.
- Examples:
  - 192.168.0.1 – Non-Routable Address
  - 127.0.0.1 – Loopback or Home Address
  - 8.8.8.8 – Common DNS Address



IPv4 – Key Concepts

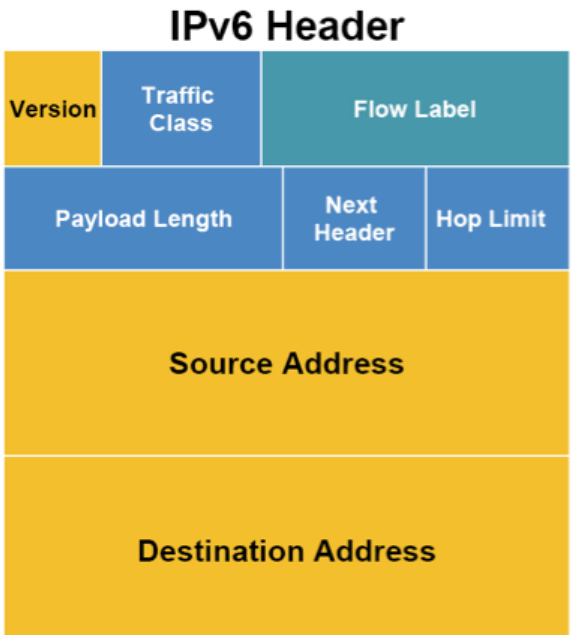
Reserved Addresses

- Two for every subnet (and cannot be assigned)
  - Network address: the first address in a subnet – 192.168.1.1
  - Broadcast address: the last address in a subnet – 192.168.255.255
- Private Address Space
  - Used on internal networks, not routable on the Internet
  - Not Internet-routable (and should be blocked at the perimeter)
  - Defined by RFC1918

Network Layer (Layer 3)

IPV6

- 8 sets of 16-bit numbers separated by a colon (:) and written in hexadecimal notation
  - Hexadecimal is base 16; each hexadecimal number represents 4 bits
  - Examples: 2001:0db8:0:130F::87C:140B = 2001:0db8:0000:130F:0000:0000:087C:140B
  - More address types: loopback, multicast, link-local unicast, site-local unicast, global unicast



Dynamic Host Configuration Protocol (DHCP)

- Provides IP address, DNS, routing, and other key configuration parameters
- Reduces administrative overhead
- Can make network forensics more difficult

IP Security Considerations Example

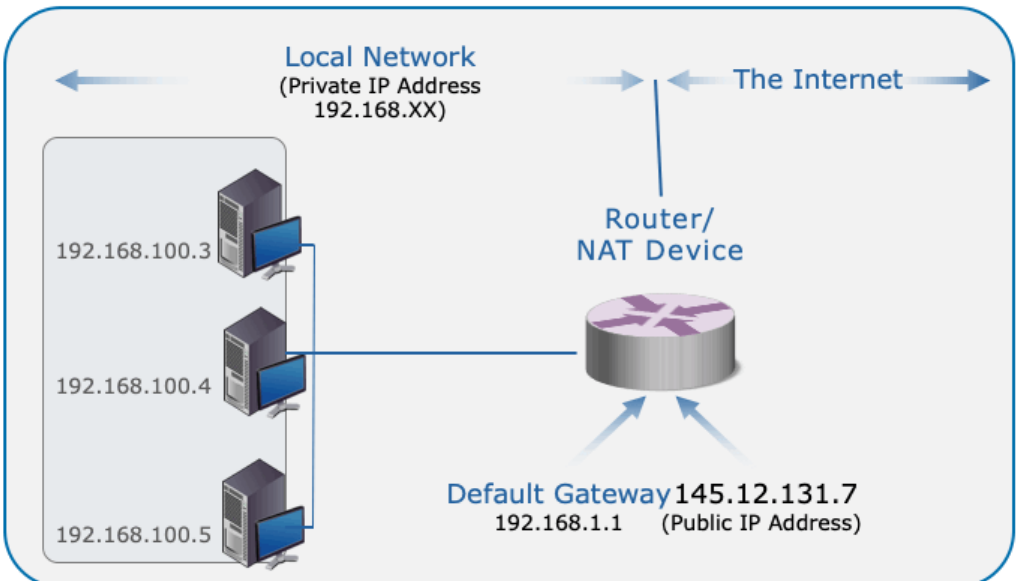
Network Address Translation (NAT)

- Provides translation of IP addresses from private address to a routable address

Port Address Translation (PAT)

- Modifies the source and destination ports

IP Routing



Transmission Control Protocol (TCP)

TCP provides reliable delivery of traffic

- Key Attributes:
  - Connection-oriented protocol
  - Reassembly of packets at destination
  - Resending of packets not acknowledged
  - Provides flow and congestion control

Related Protocols:

- File Transfer Protocol (FTP): Ports 20 and 21
- Hypertext Transfer Protocol (HTTP): Port 80
- Secure Sockets Layer (SSL): Port 443
- Secure Shell (SSH): Port 22
- Remote Desktop Protocol (RDP): Port 3389

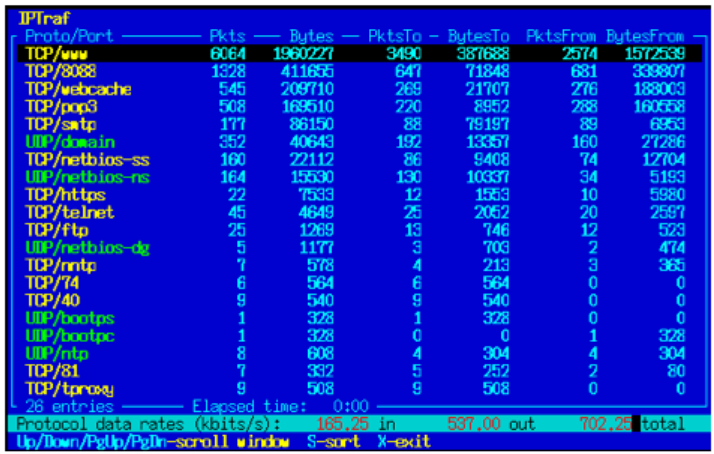
Ports

Used by TCP and UDP (and others) to map network services to host processes

- Process + Port = Socket
- 16-bit number: 1-65535

Types

- Well Known
- 0–1023
- Used by system processes to provide common network services
- Usually require root privileges to bind on the host
- Registered
- Formally assigned by the Internet Assigned Numbers Authority (IANA) for use with a certain protocols or applications
- Port numbers 1024–49151
- Dynamic
- 49152+



Note: Port mappings can be changed (which can be for normal and malicious reasons)

TCP Flags and Handshakes

TCP state is communicated via flags; most common are as follows:

ACK: Acknowledge the receipt of a packet

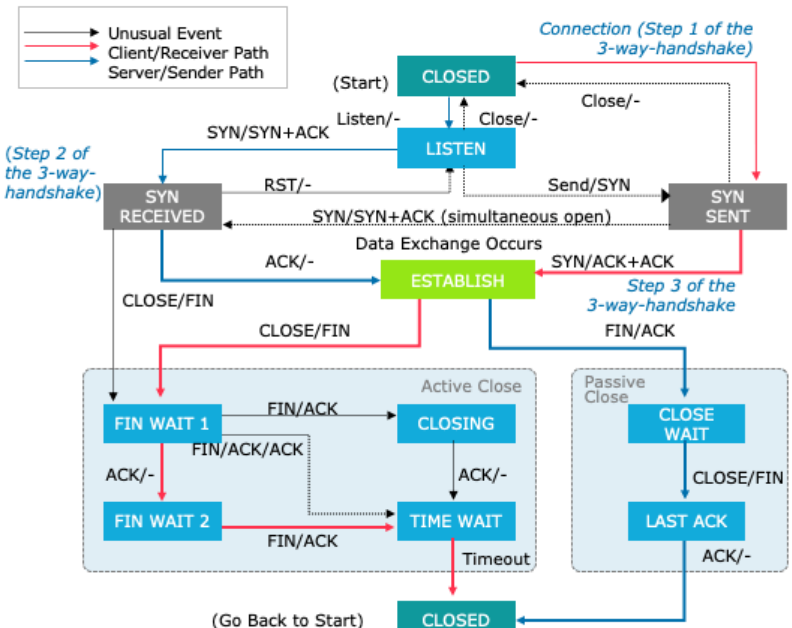
PSH: Push (send) the buffered data to the receiving application

RST: Reset the connection

SYN: Initiate a new connection by resetting the sequence numbers

FIN: No more data from sender

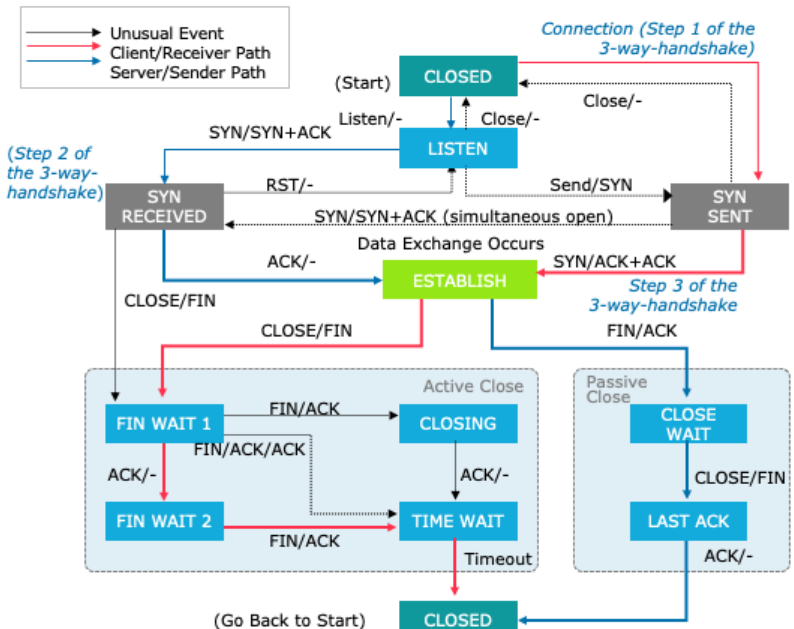
Others: Nonce Sum (NS), Congestion Window Reduced (CWR), Explicit Congestion Notification Echo (ECE), Urgent (URG)



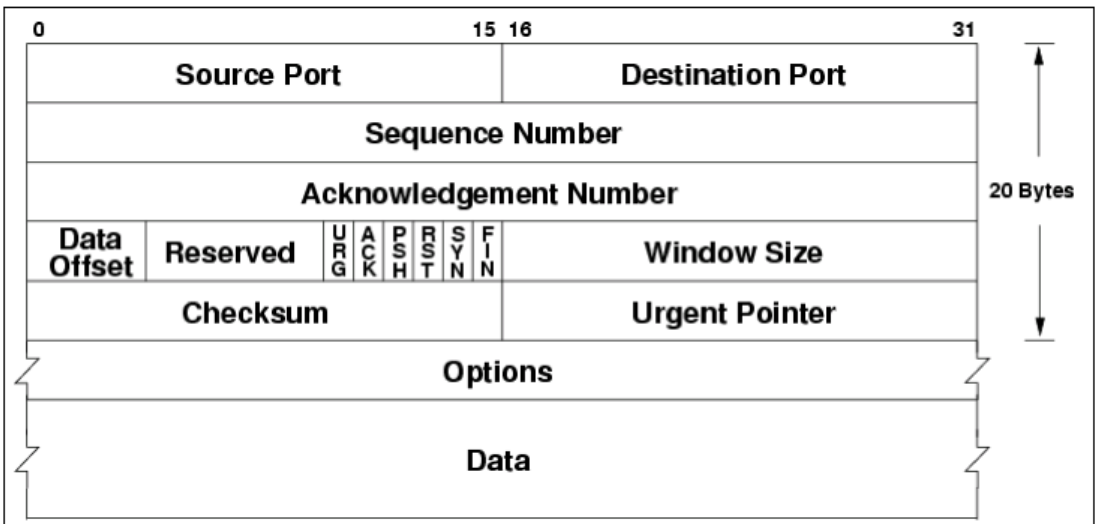
TCP Flags and Handshakes (cont.)

Normal Connection Establishment (the 3-way handshake)

- Client sends TCP packet with following characteristics:
  - SYN flag set
  - Sets sequence number to value X
- Server responds with the following:
  - SYN and ACK flags set
  - Sets acknowledgement number to X+1
  - Sets new sequence number to value Y
- Client responds with the following:
  - ACK flag set
  - Acknowledgement number set to X+1
  - Sequence number set to Y+1
- Connection established.

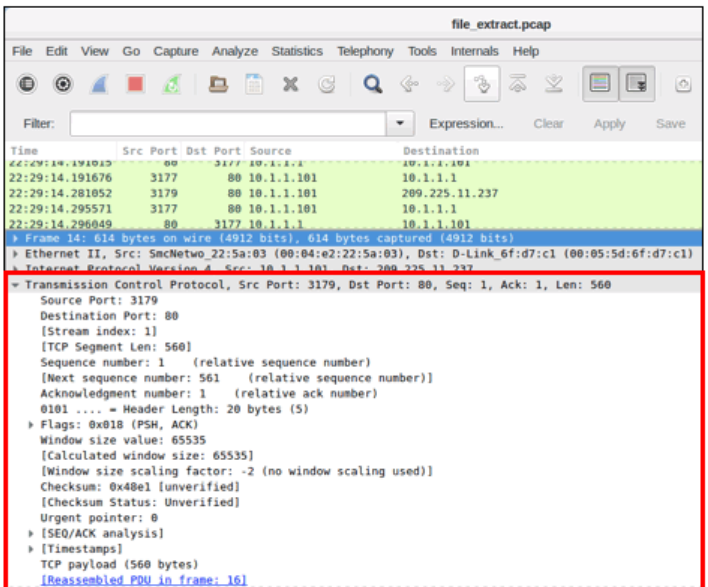


TCP Header



TCP Header Example

Single TCP Header Viewed in Wireshark:



User Datagram Protocol (UDP)

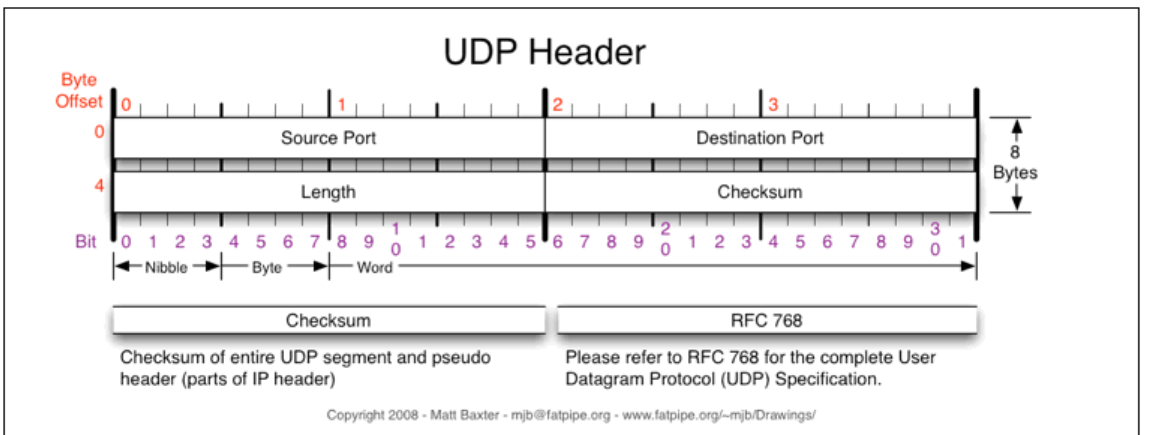
UDP provides simple, connectionless communications

- **Key attributes:**
  - Simple and lightweight
  - Unreliable – datagrams are transmitted with no protection against loss, duplication, or integrity.
  - Stateless – UDP has no concept of sessions or streams.

Related Protocols

- **Domain Name Service (DNS):** Port 53
- **Simple Network Management Protocol (SNMP):** Port 161
- **Routing Information Protocol (RIP):** Port 520
- **Dynamic Host Control Protocol (DHCP):** Port 67 to send, Port 68 to receive
- **Syslog:** Port 514

User Datagram Protocol (UDP) Header



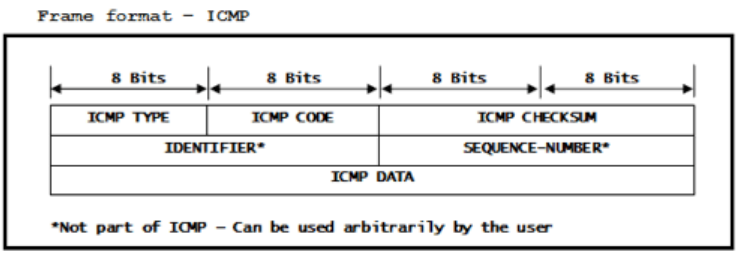
User Datagram Protocol (UDP)

TCP	UDP
Reliable	Unreliable
Connection-oriented	Connectionless
Segment retransmission and flow control through windowing	No windowing or retransmission
Segment sequencing	No sequencing
Acknowledge segments	No acknowledgement

Internet Control Message Protocol (ICMP)

Designed to communicate error and diagnostic messages

- Uses "types" and "codes"
  - Type 0 Code 0: Echo reply (ping reply)
  - Type 8 Code 0: Echo request (ping request)
  - Type 9 Code 0: Router advertisement
  - Type 3 Code 0: Destination network unreachable
  - Type 3 Code 1: Destination host unreachable



Key Protocols and Services

Protocols and services playing key roles in APT detection and mitigation:

- Exploitation by the adversary
- Detection and mitigation functions on the network
- Understanding the capabilities, normal behavior, and attack surface is critical to detecting malicious activity

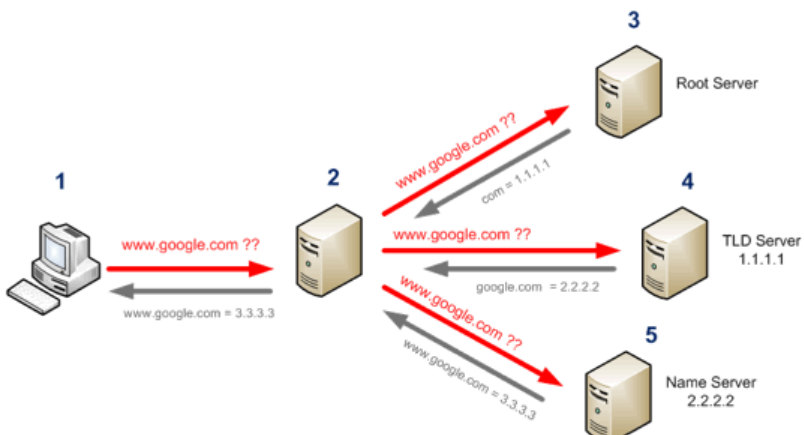
Important Protocols and Concepts:

- Domain Name Service (DNS)
- Web (HTTP, HTTPS)
- File Transfer Protocol (FTP)
- Simple Mail Transfer Protocol (SMTP)
- Obfuscation Techniques

Domain Name Service (DNS)

Translate between domain names and IP addresses

- Provide other important domain-level information
  - Authority and ownership
  - Mail servers
  - Aliases ("www.example1.com" is actually "web1.example1.com")
- Port and Protocol
  - Protocol: UDP normally; TCP for large queries and server-to-server communication such as zone transfers
  - Port: 53



Domain Name Service (DNS) (cont.)

Why is DNS important to us?

- Infected hosts commonly use DNS to communicate with external systems
- HTTPS is a common delivery method for APT
- Command and Control (C2)
- Effective detection and mitigation capabilities (DNS "black holes" with known malicious domain and IP addresses)

DNS Name Space

Suffix	Purpose	Example
com	Commercial organizations (businesses)	microsoft.com
edu	Educational organizations such as colleges and universities	berkeley.edu
gov	Governmental organizations such as the IRS, SSA, NASA, and so on	nasa.gov
mil	Military organizations	army.mil
net	Networking organizations such as ISPs	mci.net
org	Noncommercial organizations such as the IEEE standards body	ieee.org
int	International organizations such as NATO	nato.int

DNS: Common Record Types

Record	Description
A:	IPv4 address record; map hostname to IP address; also used for black holes
AAAA:	IPv6 address record
CNAME:	Alias; translates the exact domain name ("example.com" but not www.example.com) to another and continues the lookup
DNAME:	Alias with subnames; translates the domain name and all related subnames and continues the lookup with the new name
MX:	Returns list of mail transfer agents (typically external mail servers) for given
NS:	Returns the authoritative name servers for given domain
PTR:	Points to a canonical name but does not translate; typically used for reverse lookups
SOA:	Start of authority; returns the top-level nameserver, responsible individual, and other useful information for given domain
SRV:	Allows the lookup of specific services on a network (e.g., Domain Controllers on a Windows network)
TXT:	Arbitrary text data; usually blocked

DNS: Dynamic DNS (DynDNS)

Allows a network device to change IP address while maintaining the same domain name and is updated in real time

- DynDNS typically uses a client program that constantly updates the DynDNS service with the updated server IP.
- Can be procured quickly, freely, and anonymously; also easy to manage
- Botnet masters and APT adversaries regularly use DynDNS domains.
- DynDNS is difficult to protect against.

HTTP and HTTPS

Why is the web such a big issue?

- Web traffic provides a large attack surface
- Common Delivery Method for APTs
  - Drive-by download attacks
- Regularly Used for C2 (especially HTTPS)
- Port and Protocol
  - Protocol: TCP and many others
  - Port: 80 for HTTP, 443 for HTTPS
- Key Points and Security Considerations
  - Any system/application that understands text streams and network sockets can communicate via HTTP
  - Know browser-based attacks and how they work

Common Web Attacks: HTTP and HTTPS

- Drive-by downloads: Common when a user downloads a malicious file
- Object vulnerabilities: Exploit unintended behaviors in web-content object handlers (Adobe Flash, ActiveX)
- Cross-Site Scripting (XSS): Leverages errors in input sanitization to attack both the server and the users
- Most browser-based attacks rely on administrative privileges

Hypertext Transfer Protocol Secure (HTTPS)

- HTTPS is essentially HTTP over Transport Layer Security
  - HTTP traffic inside a TLS session
  - Traffic below the TCP header is encapsulated in TLS and encrypted.
- Secure Sockets Layer (SSL) is not the same as HTTPS
  - TLS is the successor to SSL
  - TLS and SSL are cryptographic protocols that encrypt individual segments of network traffic
  - Both TLS and SSL are used for more than just HTTPS such as Virtual Private Network (VPN) connections
- Attacker use of HTTPS:
  - Attackers know that outbound 443 is usually open and that security devices usually ignore the contents of HTTPS traffic
  - Other applications can be easily configured to masquerade as legitimate HTTPS traffic

Hypertext Transfer Protocol Secure (HTTPS) (cont.)

Most HTTPS-specific attacks target the Public Key Infrastructure (PKI) system. Know the fundamentals:

- Public/Private keys
- Certificate authorities
- Identity vs. integrity vs. non-repudiation
- Key exchange methods

HTTP: Hypertext Transport Protocol Message Structure

HTTP communication consists of two basic actions:

- Client Request
- Server Response

Each request and response consists of the following:

- Initial Line
- Request: Contains the request method, URI, and the protocol version
- Response: The protocol version, server's response to the client request in the form of a numeric status code (such as "404"), and a reason message (such as "Not Found")
- Headers
- Empty Line
- Body

HTTP: Request Methods

HTTP defines nine methods for the client to use when requesting a resource from the server.

Most important:

- GET: Request to download a resource from a server
  - Example: "GET http://www.msn.com/?ocid=iehp HTTP/1.1\r\n"
- POST: Submit text data in the message body to be processed by the server for a specified URI
  - Example: "POST http://maps.google.com/reviews/components HTTP/1.1\r\n"

HTTP: Server Status Codes, Reason Messages

- HTTP status codes and messages
- Server can define custom codes and messages

Code	Use
1XX	Informational
2XX	Success
3XX	Redirection
4XX	Client Error
5XX	Server Error

HTTP: Server Status Codes, Reason Messages (cont.)

HTTP Headers have a critical source of information about both the client and the server (for both attackers and investigators). The 400 series are the ones seen most often.

Code	Type	Explanation
200 OK	Informational	The request has succeeded. The meaning of a success varies depending on the HTTP method.
301 Moved Permanently	Redirection	This response code means that the URI of the requested resource has been changed, the new URI would be given in the response.
400 Bad Request	Client Error Response	This response means that server could not understand the request due to invalid syntax.
403 Forbidden	Client Error Response	The client does not have access rights to the content (i.e., they are unauthorized), so server is rejecting to give proper response. Unlike 401, the client's identity is known to the server.

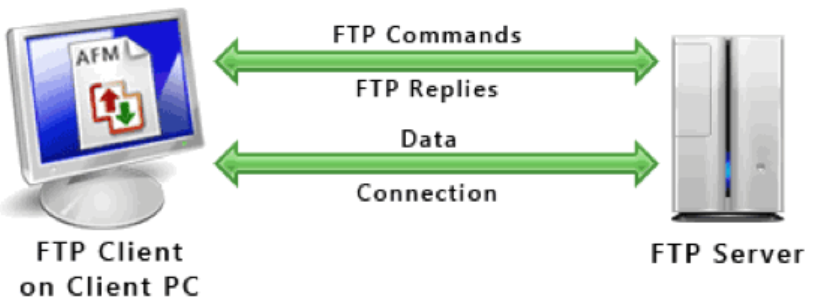
LAB008: Command Line Network Analysis

File Transfer Protocol (FTP)

Why is FTP important to us?

- Tool delivery
- Data exfiltration
- Port and Protocol
- Protocol: TCP
- Port: 20 (data), 21 (control)

- Connections
  - Control:
  - Port 21
  - Remains open for the entire sessic
  - Used to authenticate pass commands from client to server and report status from server to client using codes
  - Cleartext, ASCII based



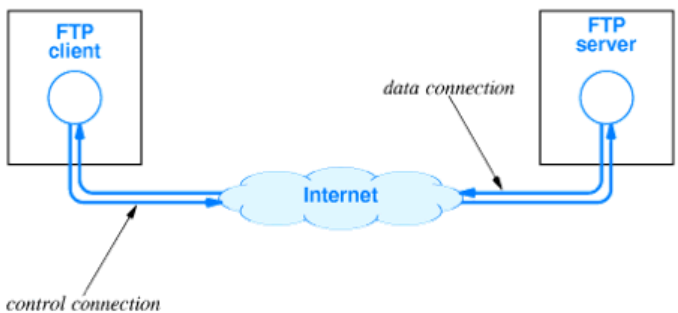
FTP: Connection and Transfer Modes

Connection Modes

- Active
  - Client indicates desire to transfer file
  - Client listens on specified port and waits for incoming server connection
  - Server initiates TCP connection to client
- Passive
  - Client indicates desire to transfer by sending "PASV" command to server
  - Server replies with IP address and port
  - Client initiates TCP connection to server

Transfer Modes

- ASCII
- Data is transferred using ASCII formatted text streams
- Will not work for binary file types
  - Binary
  - Such as "Image" types
  - Transfers data using a stream of binary data



FTP: Security Considerations

- Commonly used in businesses
- Cleartext protocol and is easily inspected
- There are multiple FTP-like encrypted transfer mechanisms:
  - FTPS: FTP over SSL (aka FTP Secure)
  - SFTP: Secure Shell (SSH) File Transfer Protocol
  - SCP: Secure Copy Protocol; also based on SSH

Email

Email is often an issue

- Email is a common vector for APT delivery

Common Types:

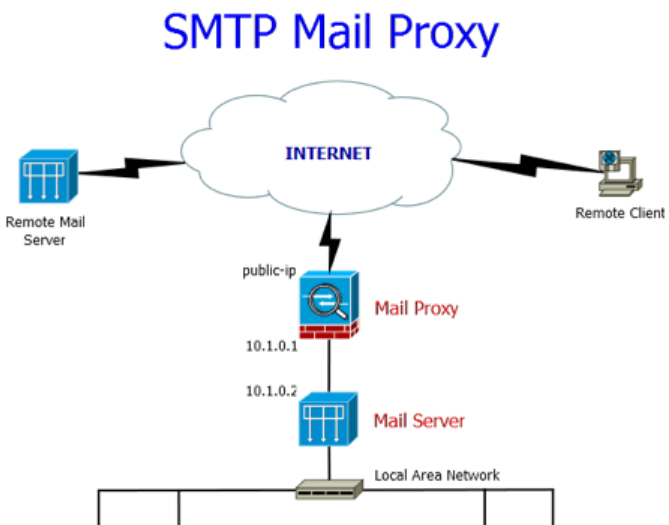
- Internet Message Format (IMF) and Simple Mail Transfer Protocol (SMTP)
- WebMail
- Microsoft Exchange and Messaging Application Programming Interface (MAPI)

SMTP - Simple Mail Transfer Protocol

SMTP is a standard for Internet email delivery. It defines the transport, not the format; SMTP provides the mail "envelope"

- Port and Protocol are as follows:
  - –Protocol: TCP
  - –Port: 25 and 587

Message routing is depicted on the slide



SMTP - Simple Mail Transfer Protocol (cont.)

Similar to HTTP

- Messages are essentially streams of ASCII text separated by carriage returns
- Contain a Header and Body
- Heavily reliant on MIME types

Message Structure

- Header:
- Pairs of field headings and values separated by ":"
- Records are separated by CRLFs
- Header block terminated by two CRLFs
- Body:
- Lines of ASCII text

SMTP - Simple Mail Transfer Protocol: Commands

Important Commands

- EHLO (Extended HELLO) or HELO (HELO) <username>: used by client to initiate a session with an SMTP server
- MAIL FROM <from address>: initiates a mail transaction in which a mail message is delivered to the server, including the sender's mailbox address
- RCPT <to address>: specifies the recipient of the message
- DATA <message>: the actual message, including all headers; upon receipt, the receiving system will add another TRACE record to the top of the message
- RSET: abort the current transaction
- VRFY <email address>: confirm that the specified user exists in the system
- QUIT: close the transmission channel

Server Responses

- 220: Ready
- 250: Requested action accepted and completed
- 252: Recipient cannot be VRFY'd
- 421: Service not available, connection closed
- 500/501: Syntax errors

SMTP - Simple Mail Transfer Protocol: Headers

Trace Headers

- Invaluable trail of message handling
- Every server in the transport process must insert trace messages at the beginning of the message.
- SMTP servers must not change or delete a received line.
- Return-Path: Added by the final system that delivers the message to the recipient
- Received: Added by each system involved in the transport of the message

Multipurpose Internet Mail Exchange (MIME)

- Internet standard initially designed to extend the format of email
  - Also heavily used elsewhere
- The de facto format standard for email
- SMTP and IMF are ASCII protocols; MIME extends the capabilities to allow other types of information
- Headers
  - Version: Currently at 1.0
  - Content-ID: Identify multi-part messages
  - Content-Type: Internet media type
- Content-Disposition: Defines the way MIME data is presented
- Two main types: "inline" and "attachment"
- Content-Transfer-Encoding: Indicates that a binary-to-text encoding mechanism has been used
- Encoded-Word: Used to identify non-ASCII header names and values

Network Protection

Capgemini

Questions?

Overview of Firewalls Types

- Packet Filtering
- Circuit-Level
- Stateful Inspection
- Application-Level
- Next-Gen Firewalls

Firewalls

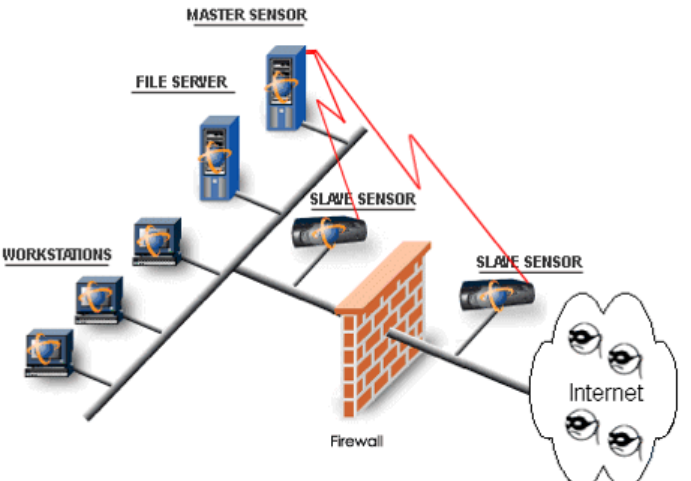
- Allow or deny traffic based on various attributes such as the following:
  - Source/Destination IP address
  - Source/Destination port numbers
  - Packet payloads
  - Encapsulated protocols/protocol inspection
- Deployed to partition network segments to provide enclaves that are protected from one another
- For C2, pay attention to the egress rules (traffic leaving the network)

Web: Proxy Servers

- Traditional network perimeter devices (routers, firewalls) operated at Layer 2 and Layer 3
- Proxy servers provide multiple useful capabilities:
  - Filtering
  - Caching
  - DNS Proxy
  - Logging and Monitoring
- Advanced functionality

Intrusion Detection Systems (IDS)/ Intrusion Prevention Systems (IPS)

- Active and Passive IDS
- Network Intrusion Detection Systems (NIDS) and Host Intrusion Detection Systems (HIDS)
- Knowledge-based (signature-based) IDS and behavior-based (anomaly-based) IDS



Questions?

LAB009: Malware Traffic Analysis

Questions?

People matter, results count.

About CapGemini

A global leader in consulting, technology services and digital transformation, CapGemini is at the forefront of innovation to address the entire breadth of clients' opportunities in the evolving world of cloud, digital and platforms. Building on its strong 50-year heritage and deep industry-specific expertise, CapGemini enables organizations to realize their business ambitions through an array of services from strategy to operations. CapGemini is driven by the conviction that the business value of technology comes from and through people. It is a multicultural company of 200,000 team members in over 40 countries. The Group reported 2017 global revenues of EUR 12.8 billion.

Learn more about us at

www.capgemini.com