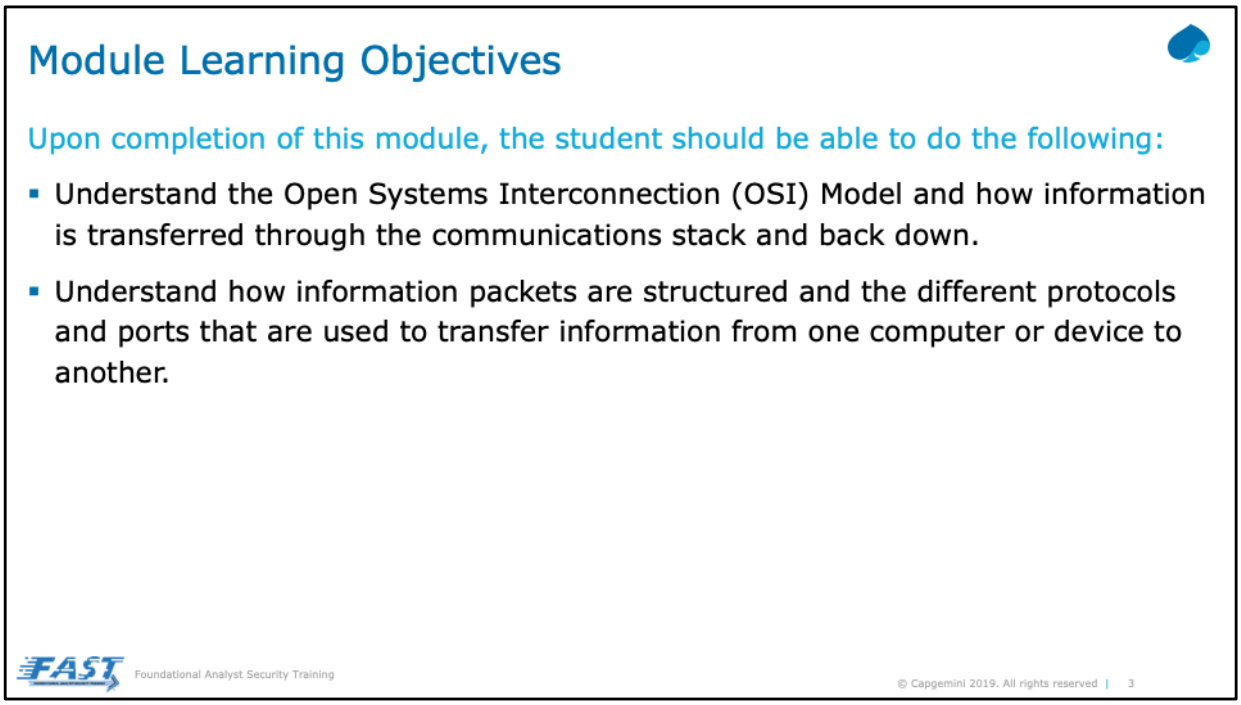

FAST
FOUNDATIONAL ANALYST SECURITY TRAINING
TM
Capgemini
Foundational Analyst Security Training
© Capgemini 2019. All rights reserved | 1

Module 6 – Networking

Capgemini

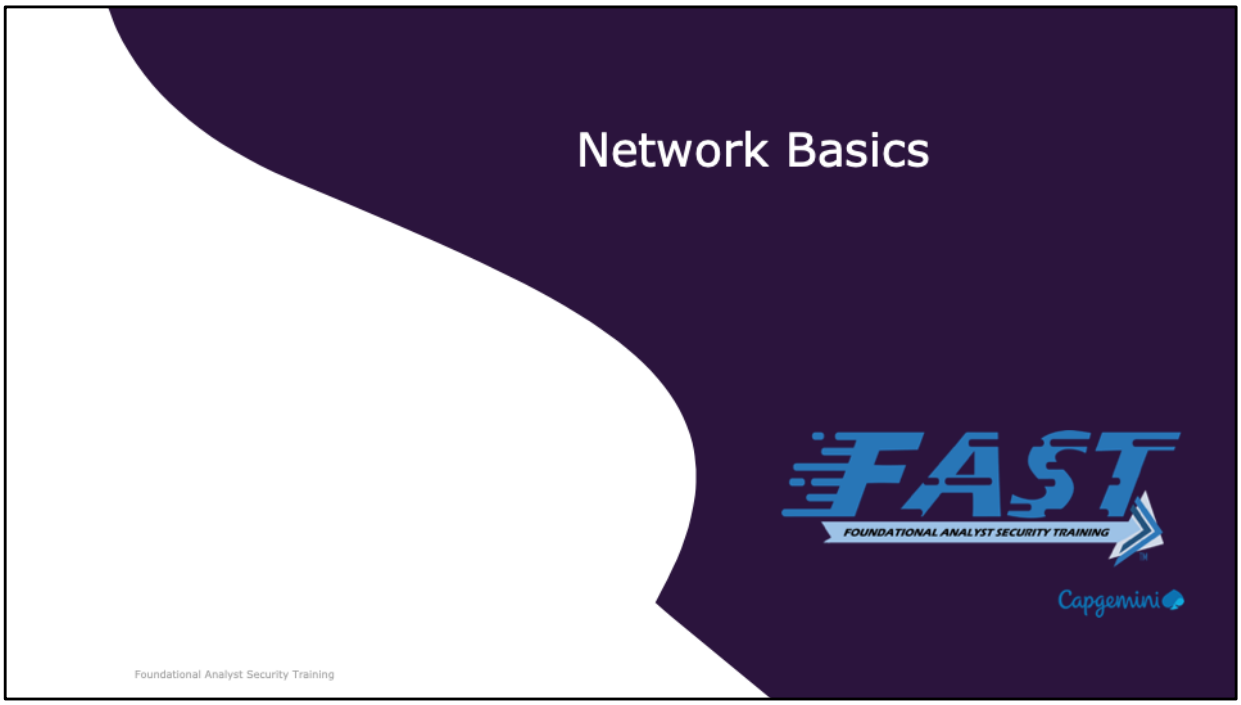


Upon completion of this module the student should be able too:

Understand the 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.

Understand how cryptography, encryption, and encoding are used to obfuscate information.



This modules covers a network refresher of many core networking and protocol topics.

Agenda

OPEN SYSTEMS INTERCONNECTION (OSI) MODEL | ROUTING

- OSI Model
- TCP/IP Model
- OSI Layers

Topic Learning Objectives

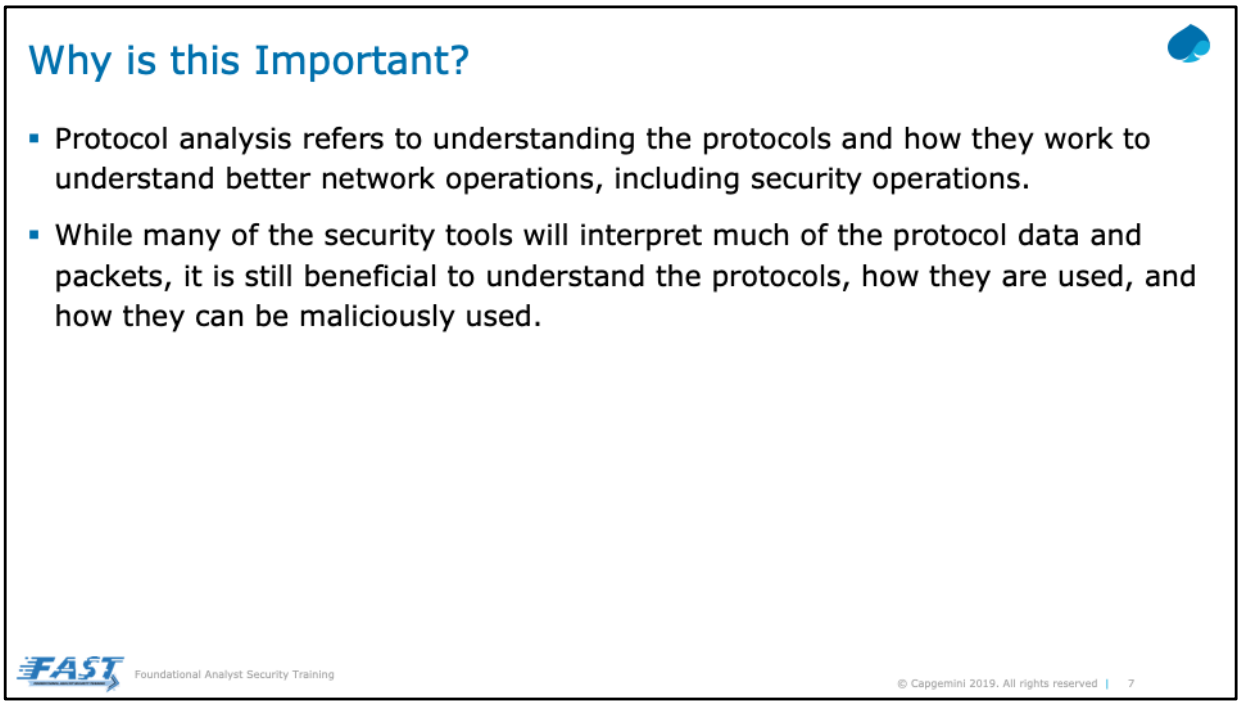
Upon completion of this topic, the student should be able to do the following:

- 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.

FAST Foundational Analyst Security Training

 6

Upon successful completion of this module, students will be able to:

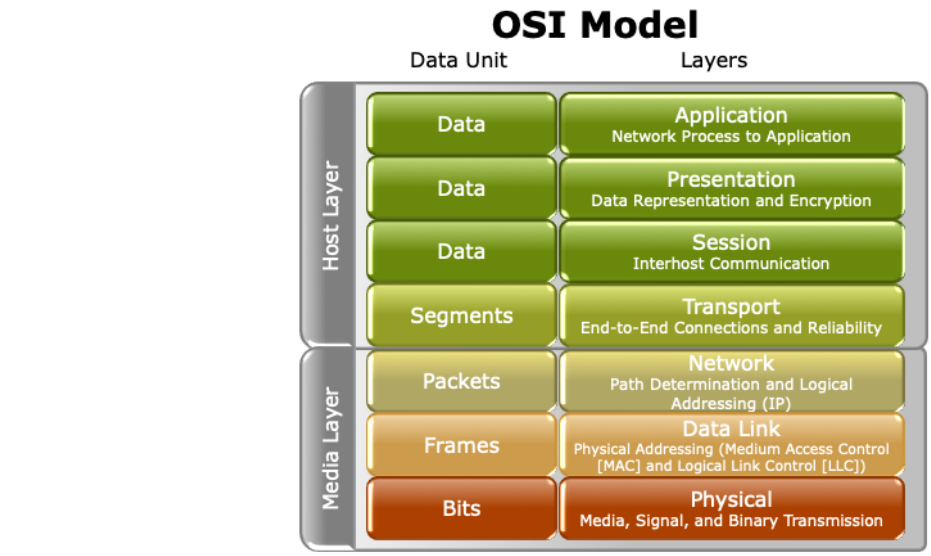


It is important to have an understanding of network and protocol concepts.

Protocol Analysis refers to understanding the protocols and how they work to better understand network operations, including security operations.

While many of the security tools will interpret much of the protocol data and packets for you, it is still beneficial to understand the protocols, how they are used, and how they can be maliciously used.

OSI Reference Model

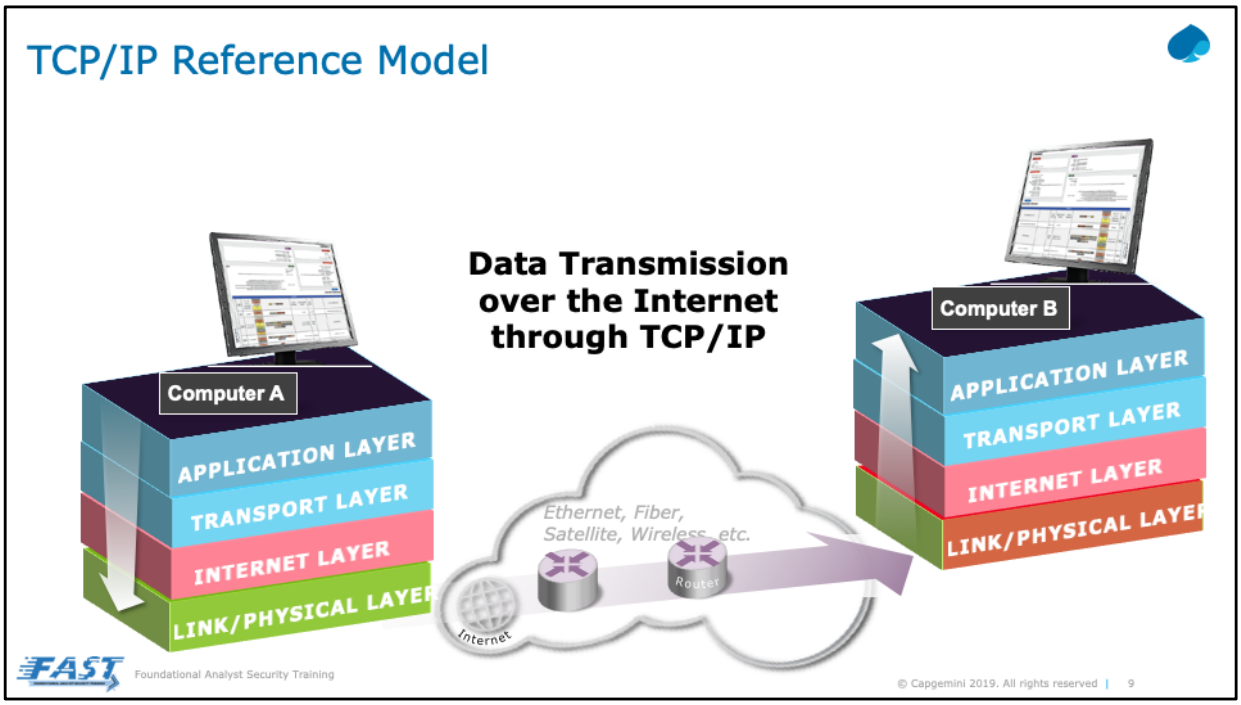


FAST Foundational Analyst Security Training

The OSI reference model is the most commonly mentioned network reference model.

The OSI reference model has 7 Layers:

- Application
- Presentation
- Session
- Transport
- Network
- Data Link
- Physical



The TCP/IP reference model is another commonly mentioned reference model with 4 layers:

- Application
- Transport
- Internet
- Subnet/Link

The sending device starts at the Application Layer, data is encapsulated down each layer to the Subnet/Link layer, then sent across the network. As data goes from one layer to another, the data is encapsulated with header and/or footer information.

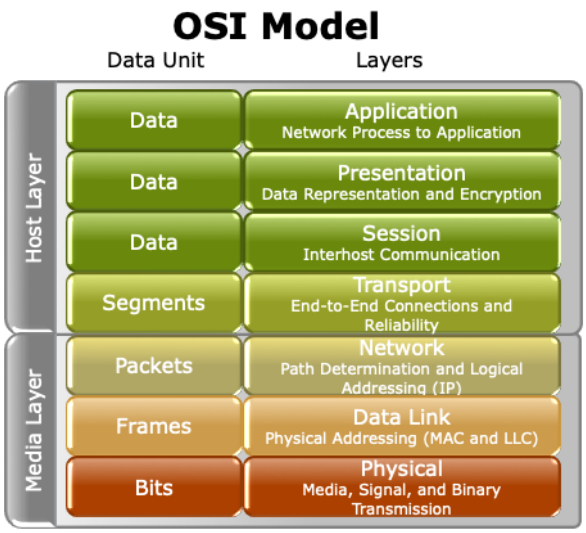
Internetworking devices, such as routers, typically look into the packets up to the Internet Layer to determine how to route the traffic, which would correspond to the Physical, Data Link, and Network Layers of the OSI reference model.

The receiving device will receive the packet(s) at the Subnet/Link Layer and pass it up the layers to the Application Layer.

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.



FAST Foundational Analyst Security Training

At the Data Link Layer (Layer 2), there are two Sublayers:

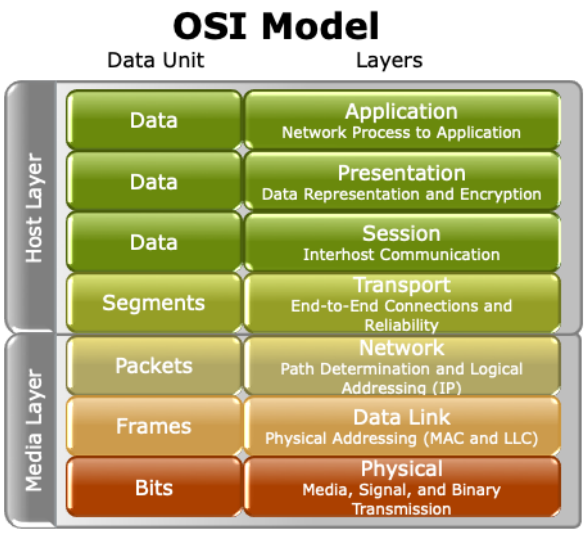
- Medium Access Control (MAC)
- Logical Link Control (LLC

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)



FAST Foundational Analyst Security Training

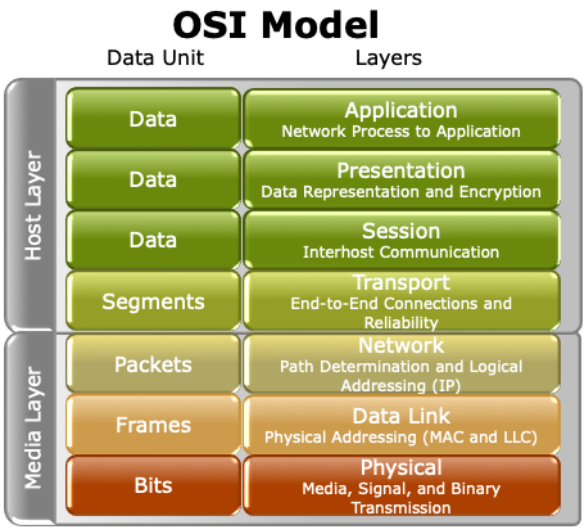
At the Data Link Layer (Layer 2), there are two Sublayers:

- 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



FAST Foundational Analyst Security Training

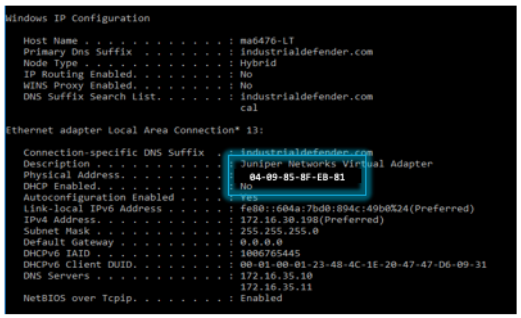
Common devices operating at this layer include:

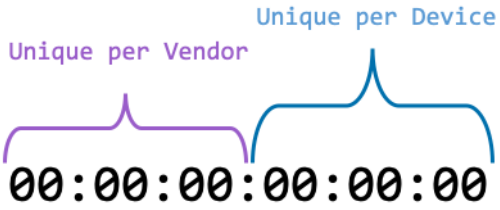
- HOSTS, which use Network Interface Cards (NICs).
- HUBS, which connect multiple hosts together. Also, hubs share the network bandwidth.
- SWITCHES, which connect multiple hosts together, have dedicated network bandwidth, and also learn MAC addresses seen on ports and build a local MAC address table.

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.



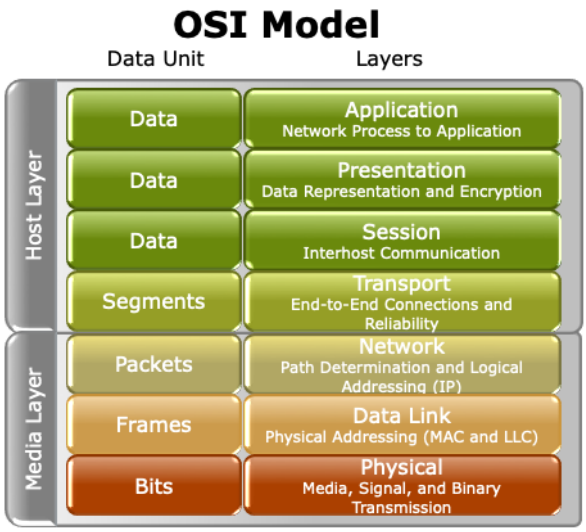


Important Data Link Layer Protocols include:
MAC addresses, or Ethernet addresses, which are 48 bits long. The first 24 bits are unique for every vendor. And, the second 24 bits are unique for every vendor's device.
Address Resolution Protocol (ARP) is a Layer 2 protocol that lets devices query the network for the MAC address of a specific IP address. Hosts devices use ARP to build a local ARP table, a cache of IP to MAC mappings.

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.



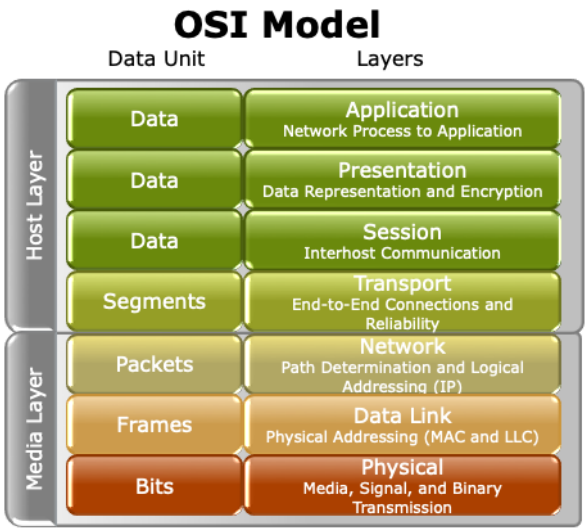
FAST Foundational Analyst Security Training

Important Data Link Layer Protocols include:
MAC addresses, or Ethernet addresses, which are 48 bits long. The first 24 bits are unique for every vendor. And, the second 24 bits are unique for every vendor's device. For example: 60:33:4B:01:23:45, or a MAC address: 60:33:4B with Vendor: Apple.

Address Resolution Protocol (ARP) is a Layer 2 protocol that lets devices query the network for the MAC address of a specific IP address. Hosts 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



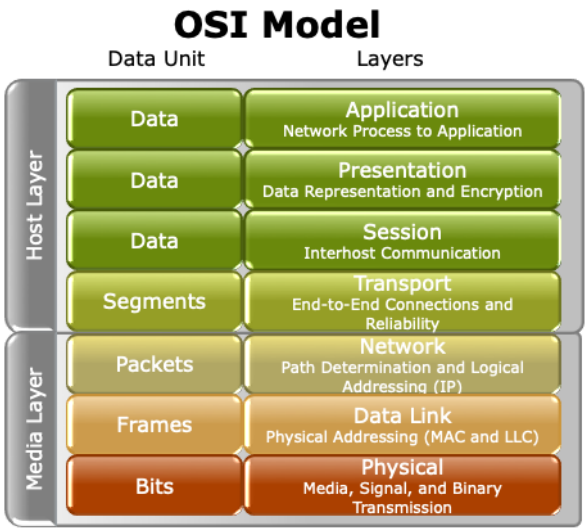
FAST Foundational Analyst Security Training

OSI Layer 3, which is the Network layer, has addresses used to route traffic between networks. The Internet Protocol (IP) is at the Network Layer. IPv4 – 32 bit addresses and IPv6 – 128 bit addresses are at this layer. IPv4 and IPv6 are covered in more detail on the next slide.

Common Network layer devices include Routers , which connect multiple subnets together.

Network Layer (Layer 3) (cont.)

- Devices
  - Routers
  - Connect multiple subnets together



FAST Foundational Analyst Security Training

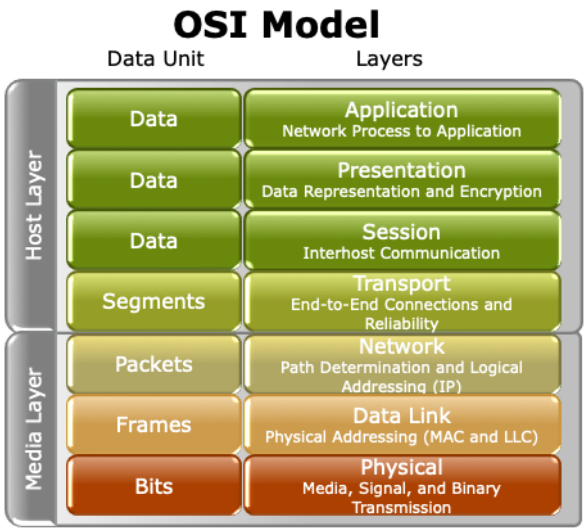
OSI Layer 3, which is the Network layer, has addresses used to route traffic between networks. The Internet Protocol (IP) is at the Network Layer. IPv4 – 32 bit addresses and IPv6 – 128 bit addresses are at this layer. IPv4 and IPv6 are covered in more detail on the next slide.

Common Network layer devices include Routers , which 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)



FAST Foundational Analyst Security Training

The Transport Layer is responsible for end-to-end communication over a network. It provides logical communication between application processes on different hosts. It also provides a point-to-point connection rather than one hop to another hop (or router).

Connections can be connection-oriented or connectionless.

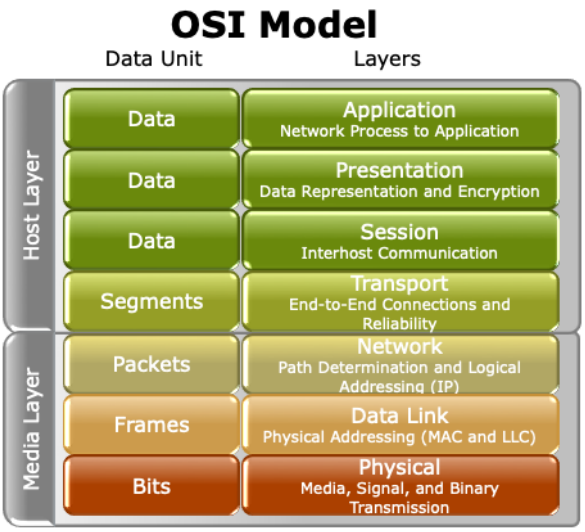
Connection-Oriented Services, such as the transmission control protocol (TCP), is used for applications in which reliable connections between hosts are necessary. TCP checks for transmission errors, lost packets, packets out of order, and other things to provide the 'reliable' connection-oriented service.

Connectionless Services, such as the user datagram protocol (UDP), just sends packets and forgets about them, providing no sessions or flow control (the advantage is speed).

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



FAST Foundational Analyst Security Training

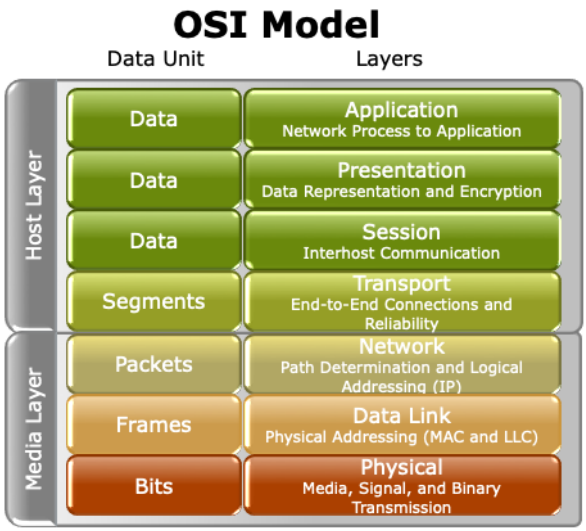
The session layer is responsible for opening and closing sessions between end-user application processes with the following characteristics:

- Controls single or multiple application connections
- Sessions are commonly implemented on Web Browsers
- Creates procedures for checkpointing, 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



FAST Foundational Analyst Security Training

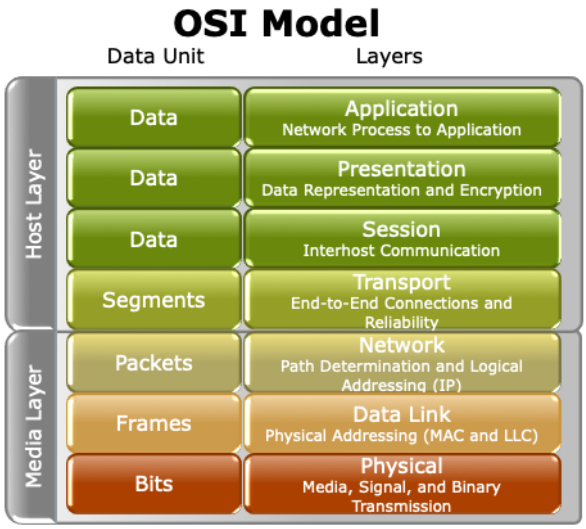
The Presentation Layer is responsible for the delivery and formatting of information, with the following characteristics:

- 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



FAST Foundational Analyst Security Training

The Application Layer provides services to the application programs, with the following characteristics:

- 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

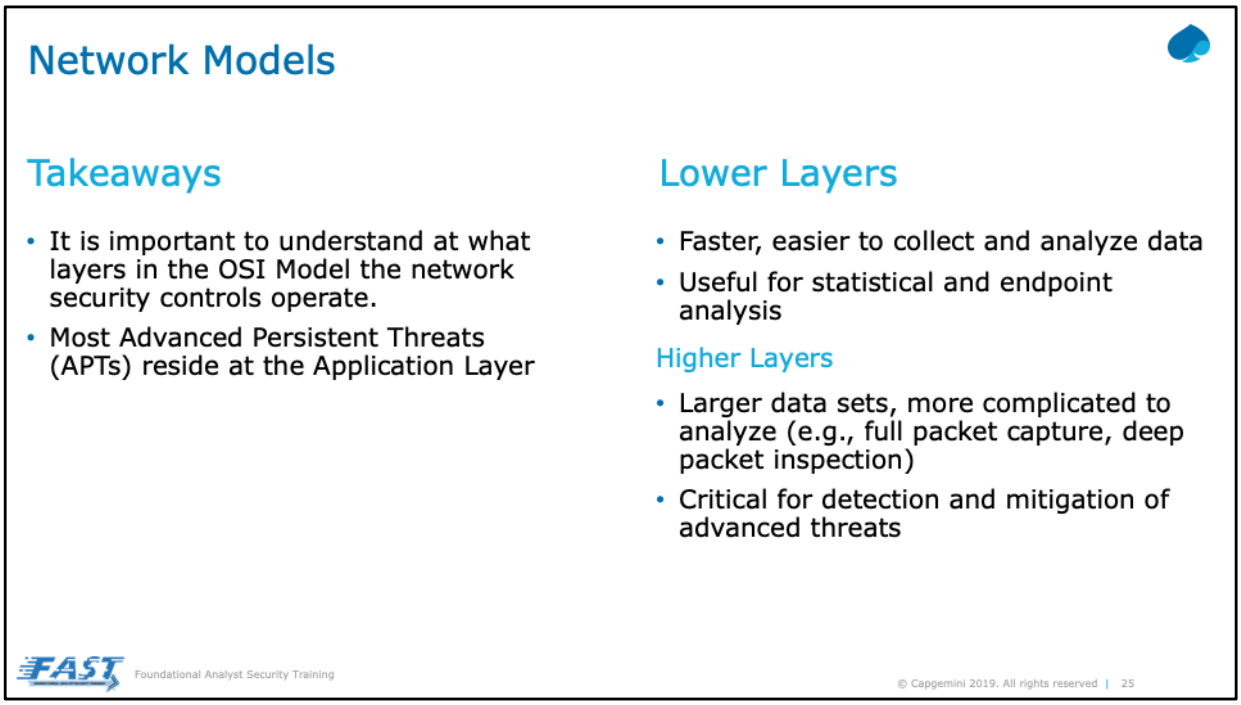
Protocol	Type	Location
Routing Information Protocol (RIP) v1 and v2	Distance Vector	Local Area Networks (LANs)
Open Shortest Path First (OSPF)	Link State	LANs, Small Wide Area Networks (WANs)
Interior Gateway Routing Protocol (IGRP)	Distance Vector	LANs; Cisco Proprietary
Enhanced Interior Gateway Routing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

FAST Foundational Analyst Security Training

Routers forward IP traffic between subnets or between networks. They use a routing table to determine the next hop or port to send the data out. There are two basic types of IP routing:

- Static: Entries in the routing table and pre-defined
- Dynamic: Dynamically reconfigure the routing table to find the "best" route

Depicted on the slide are some of the common routing protocols.

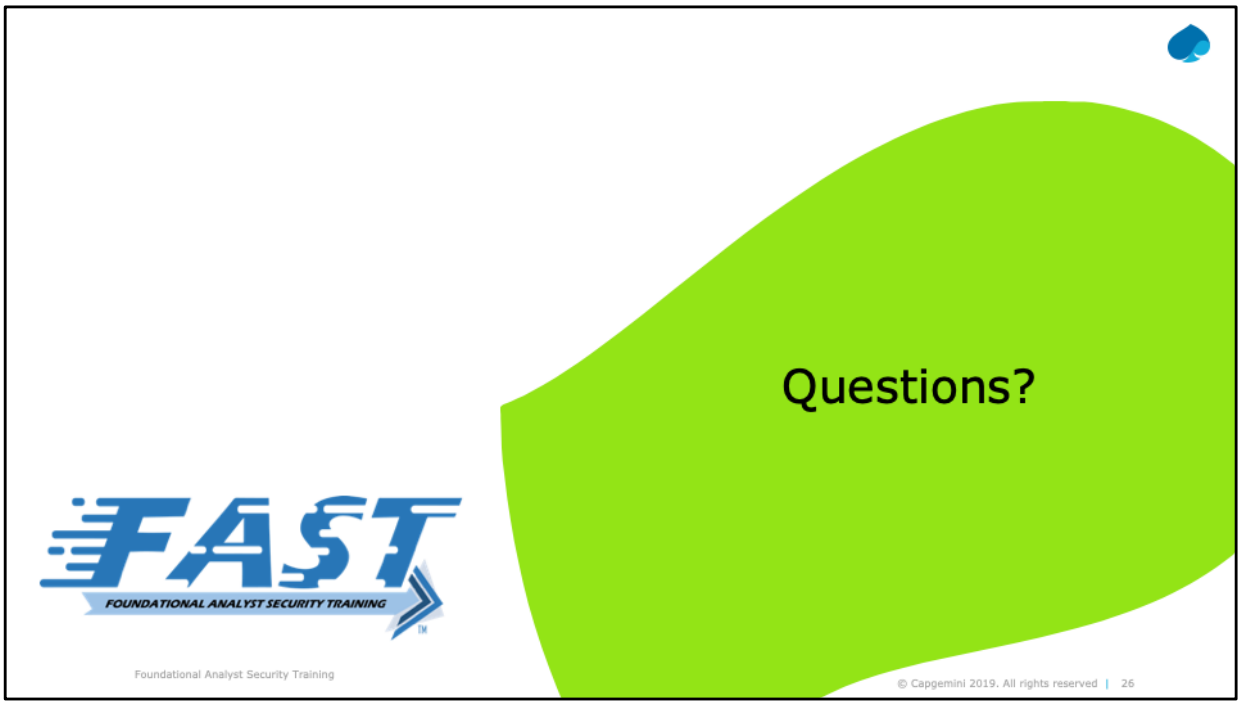


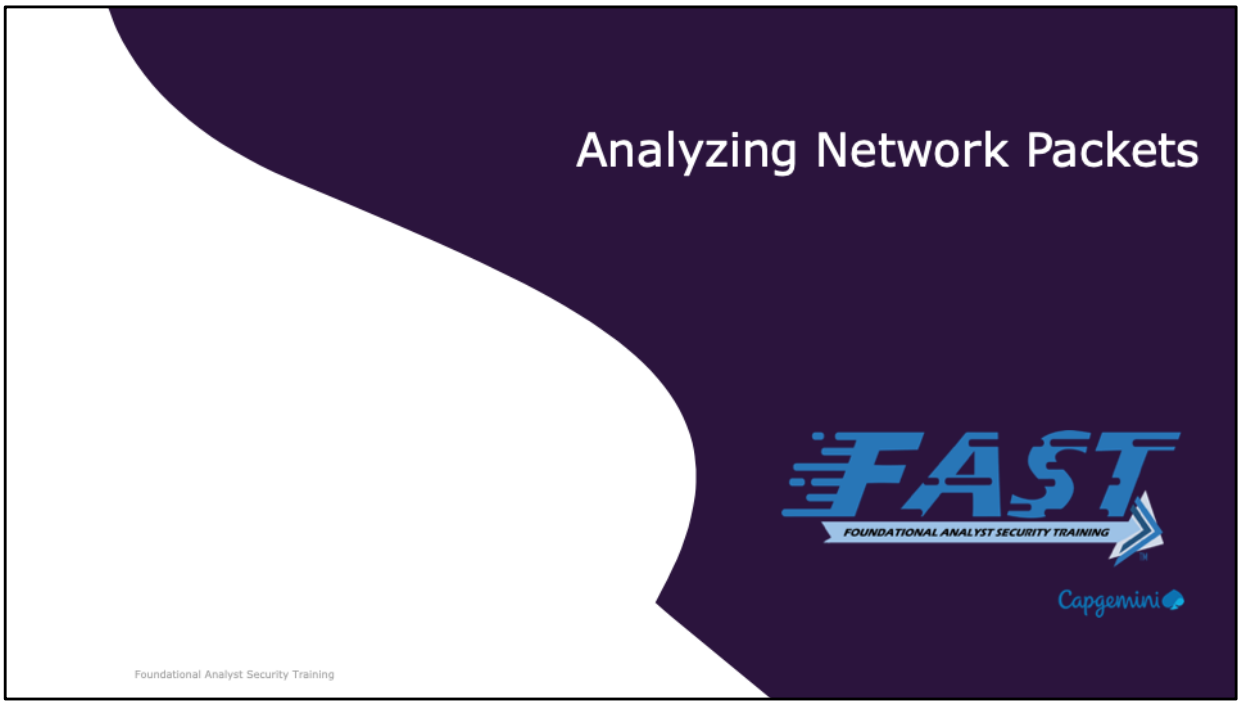
The main takeaway from this brief introduction to network models and protocols, is that it is important to understand at what layers in the OSI or TCP/IP models that the network security controls operate.

Most APTs reside at the Application layer.

Lower layer characteristics include that they are faster, easier to collect, and analyze data; and they are useful for statistical and endpoint analysis.

Higher layer characteristics include that they are larger, more complicated to analyze (e.g., full packet capture, deep packet inspection); and they are critical for detection and mitigation of advanced threats.


Questions?
FAST
FOUNDATIONAL ANALYST SECURITY TRAINING
Foundational Analyst Security Training
© Capgemini 2019. All rights reserved | 26



This module will cover Packet Analysis and methods of analyzing the data that crosses the network.

Topics covered in this module include:

- Packet Analysis Overview
- Wireshark
- Tcpdump
- Tshark
- Tcpflow
- ngrep
- Netflow
- Snort

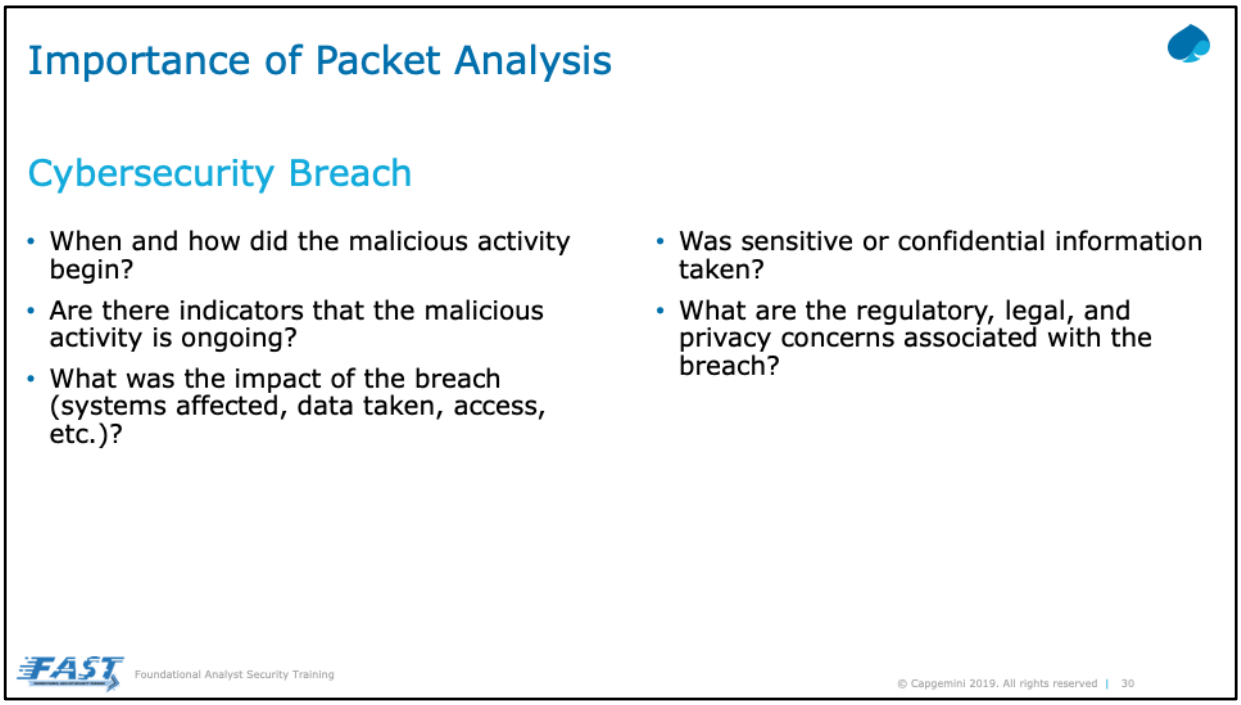
Topic Learning Objectives

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.

FAST Foundational Analyst Security Training

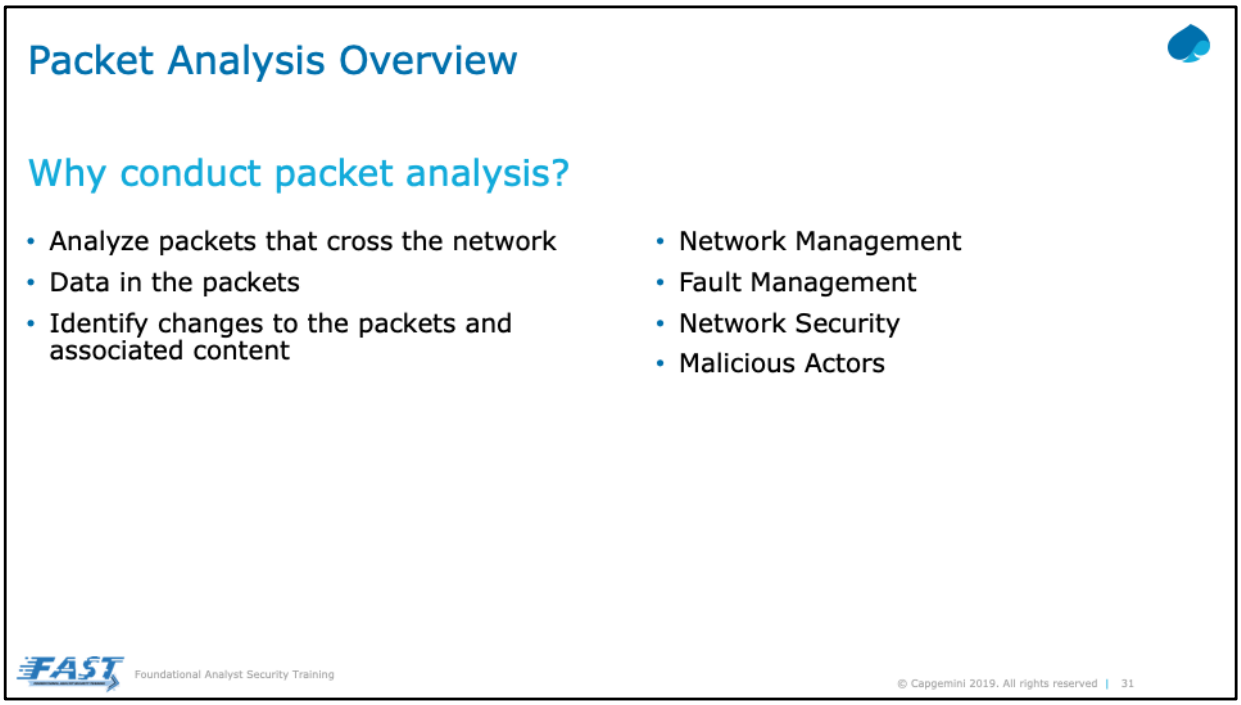
Upon successful completion of this module, students will be able to:



Consider that your organization has been breached. You have a limited amount of time to determine your response, as time becomes of the essence. You need to respond quickly from detection to containment. Organizations need to quickly and efficiently determine various factors, such as:

- 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

There are many relevant questions that needs to be answered quickly and efficiently. Thus, the importance of network forensics and packet analysis activities are paramount.

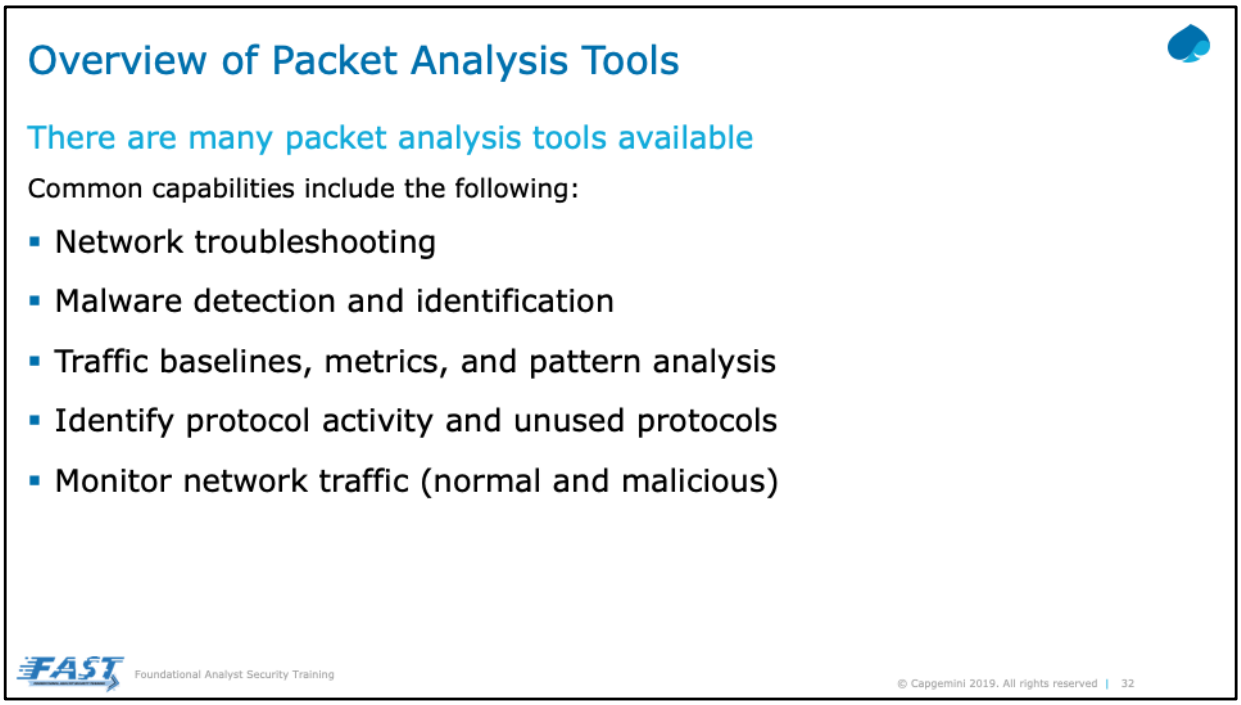


You will often need to analyze packets that cross the network, and as mentioned previously, there are many tools available to aid in the process of analyzing these packets.

You will need to analyze the data in the packets and identify changes to the packets and their associated content. These changes will often be indicators to the type of attack that is being performed or the desired goal of the attack.

Packet analysis is not just a cyber security activity, there are many reasons why you might conduct a Packet Analysis, such as:

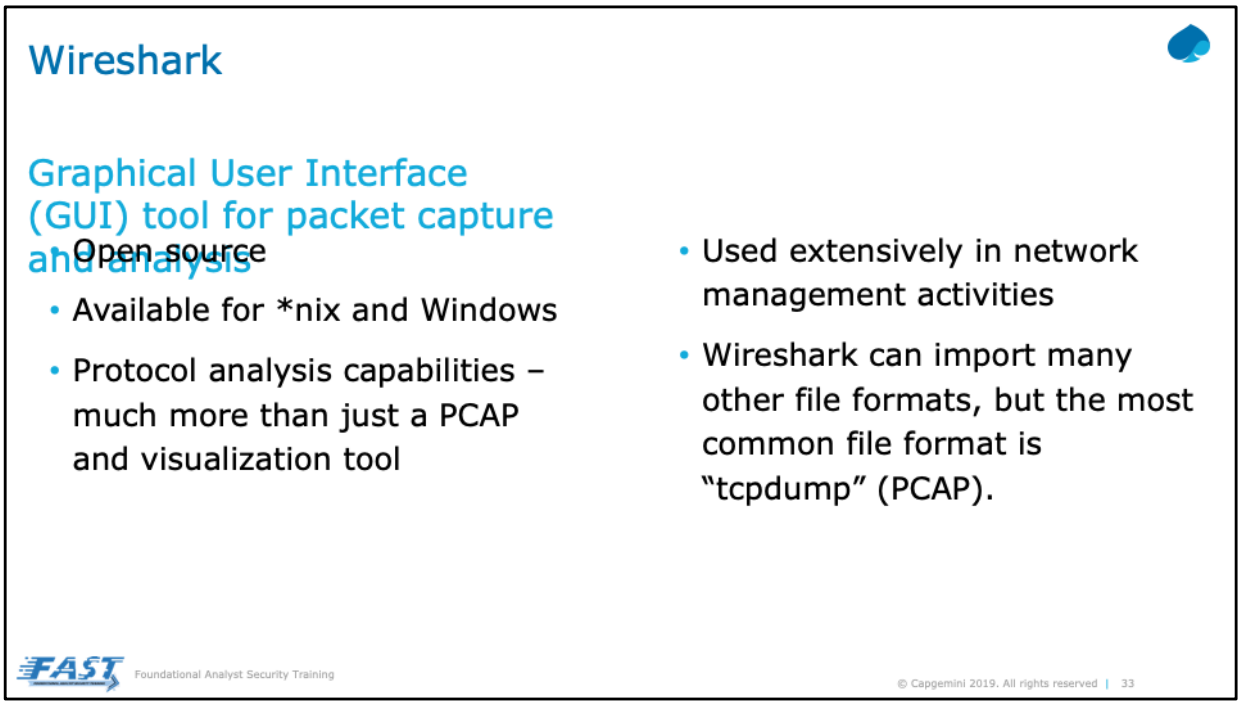
- Typical network management activities, such as performance and fault management activities.
- When trying to determine the root cause of a network issue during fault management.
- During network security activities, which is what is focused on in this course.
- Malicious actors will also use network forensics and packet analysis activities extensively when planning and conducting their attacks.



There are many packet analysis tools available on the market today; likewise, these tools can be customized and customized tools can be developed.

Common packet analysis tool capabilities include:

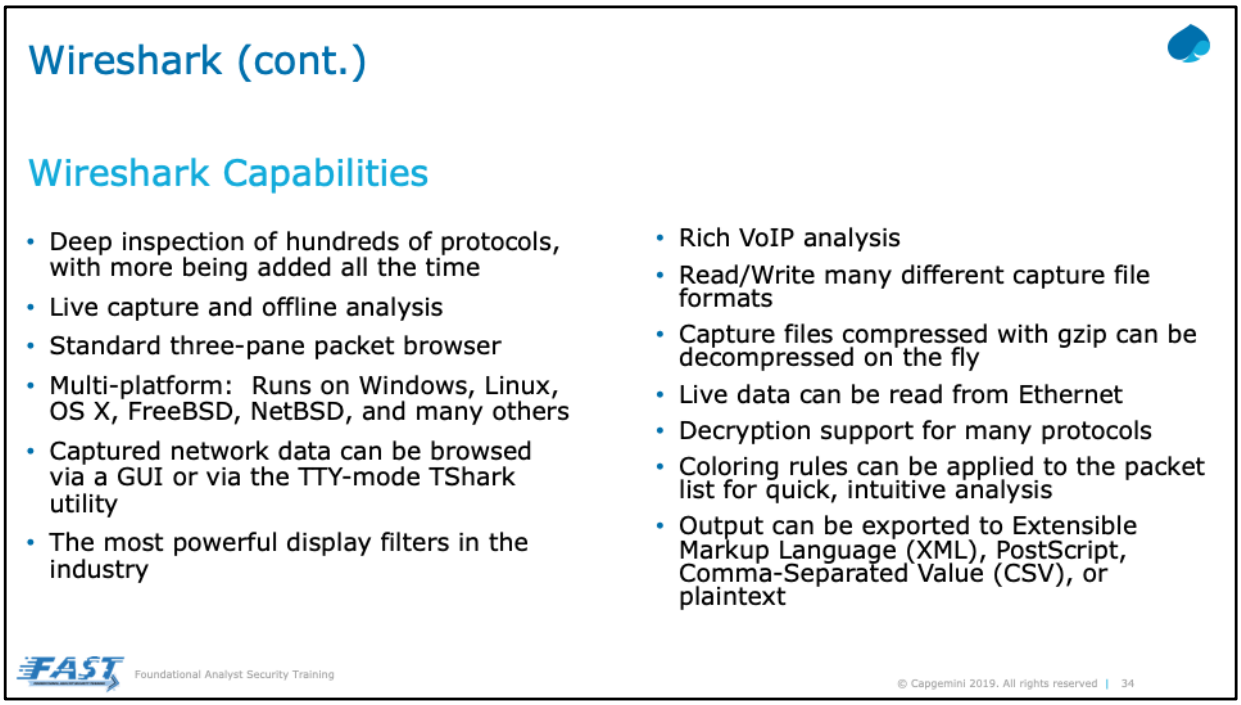
- Network troubleshooting
- Malware detection and identification
- Traffic baselines, metrics, and pattern analysis
- Identify protocol activity and unused protocols
- Monitor network traffic (normal and malicious)



Update Image

Wireshark is a popular Graphical User Interface (GUI) tool for packet capture and analysis. It is Open Source and available for *nix and Windows systems. Wireshark provides protocol analysis capabilities; it provides much more than just a packet capture and visualization tool, as it is also used extensively in other areas, such as network management activities.

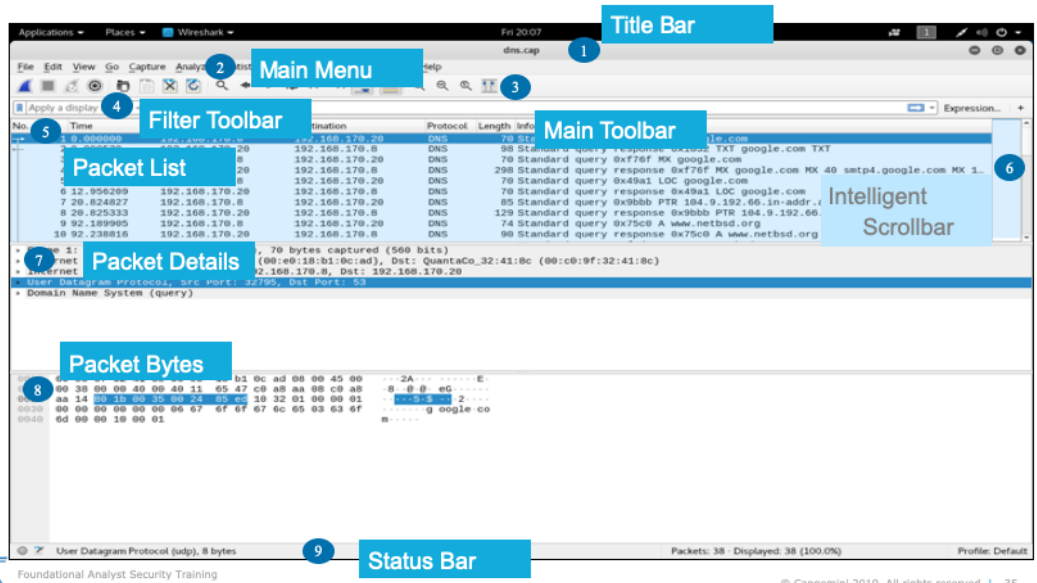
Wireshark can import many other file formats, but the most common file format is “tcpdump” (PCAP).



Wireshark is a very popular tool and has many capabilities including (as advertised by Wireshark, at https://www.wireshark.org/about.html):

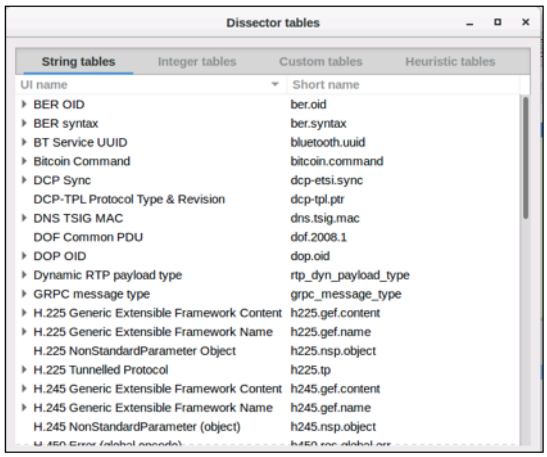
- Deep inspection of hundreds of protocols, with more being added all the time
- Live capture and offline analysis
- Standard three-pane packet browser
- Multi-platform: Runs on Windows, Linux, OS X, FreeBSD, NetBSD, and many others
- Captured network data can be browsed via a GUI, or via the TTY-mode 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 XML, PostScript, CSV, or plain text

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.



FAST Foundational Analyst Security Training

Update Image

Wireshark understands protocol formats via hundreds of protocol dissectors that enable Wireshark to parse bits in each packet.

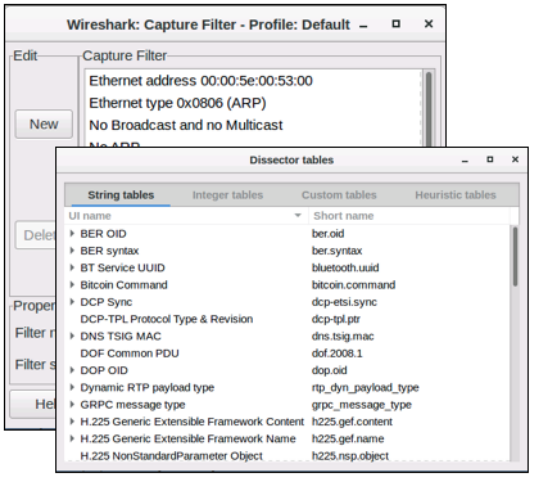
Wireshark interprets the dissector for each packet based on static assignments (defined by the user), port number, and other factors. Each field can then be used to filter the results.

port numbers - Note that it determines protocol based on the port number not the traffic itself. Port 443 is ssl traffic and if you filter on ssl traffic you will get all traffic over port 443 regardless of if the traffic is really ssl traffic.

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



FAST Foundational Analyst Security Training

Wireshark has two types of filters: Capture and Display.

Capture filters are used in the Wireshark Capture Options screen (see Wireshark's reference material for more details).

Capture Filter examples are:
To capture only traffic to or from IP address 122.118.5.4: host 122.118.5.4

To capture traffic to or from a range of IP addresses: net 210.118.0.0/24

To capture traffic from a range of IP addresses: src net 210.118.0.0/24

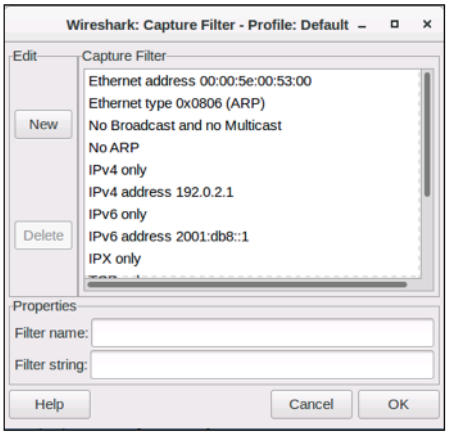
To capture traffic to a range of IP addresses: dst net 210.118.0.0/24

To 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$"



The other type of filter are Display filters, which are from the Main display screen.

Display filter examples are:

To show only SMTP (port 25) and ICMP traffic: tcp.port eq 25 or icmp

To show only traffic in the 210.118.0.0/16 subnet: ip.src==210.118.0.0/16 and ip.dst==210.118.0.0/16

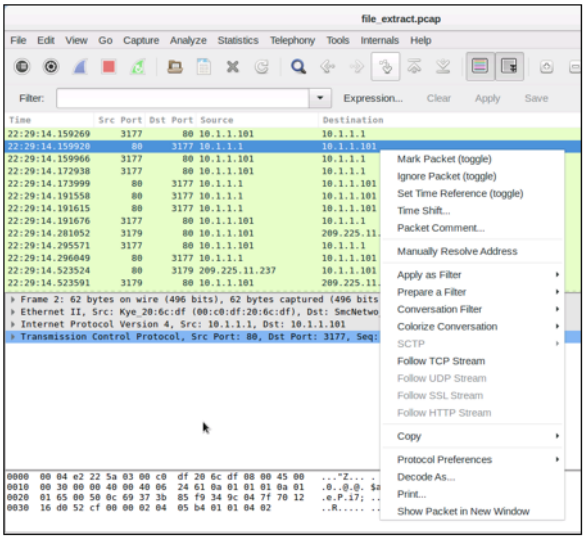
To match HTTP requests for PHP pages: http.request.uri matches "php$“

when you use follow stream it is setting a filter, and you will only see data associated with the stream in Wireshark. If you search for something not in the stream you will not get any hits. So as with filters you need to remove filters before searching.

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.



FAST Foundational Analyst Security Training

 | 39

Update image

This slide is depicting the pop-up to select Filter Field Reference.

Right-click on the Protocol Filter to display the pop-up. Choose "Filter Field Reference" to see the fields defined for the selected protocol.

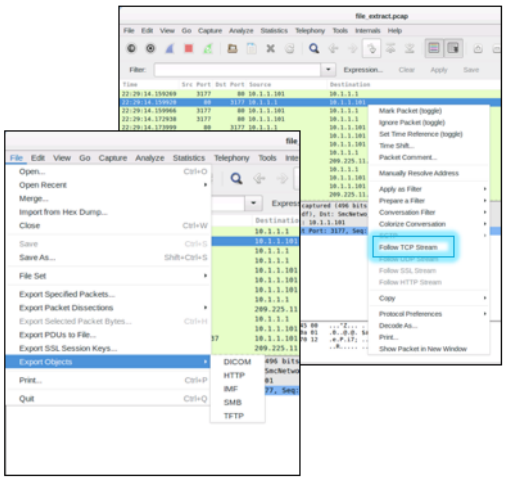
Protocol Dissectors include:

- Frame
- Ethernet
- IPv4
- TCP
- HTTP

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



FAST Foundational Analyst Security Training

Update image

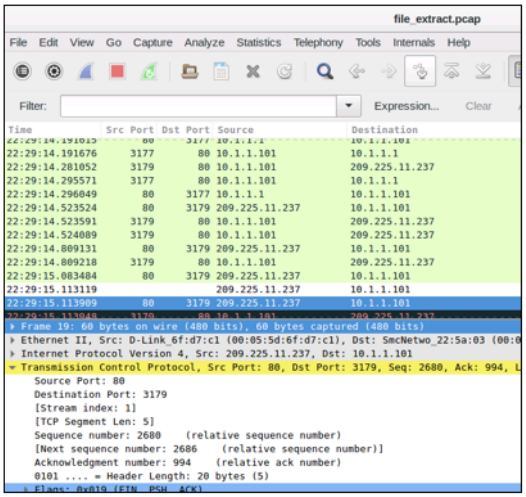
A very useful feature of Wireshark is the ability to rebuild data streams. Instead of viewing groups of packets, Wireshark will reconstruct the raw data and display it in the order that the application sees it.

Right-click on a packet and choose "Follow TCP Stream".

Wireshark Experts

Rebuilding Data Streams and Extracting Files

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

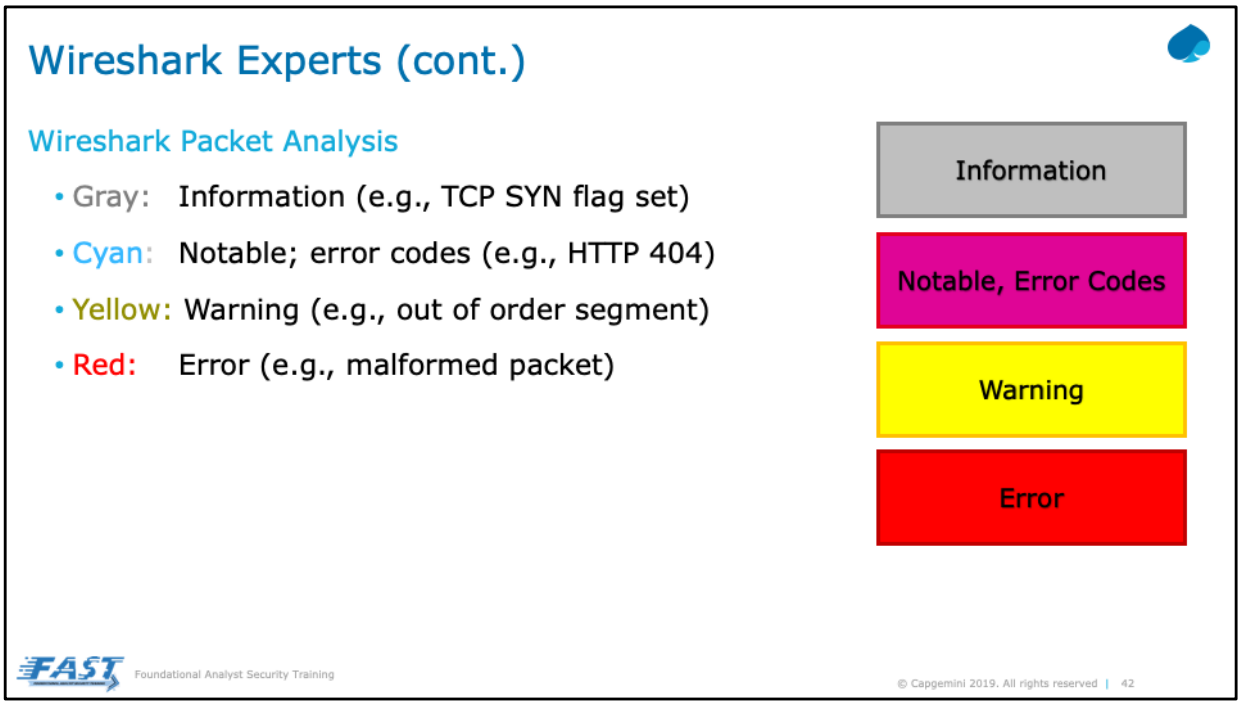


FAST Foundational Analyst Security Training

Update image

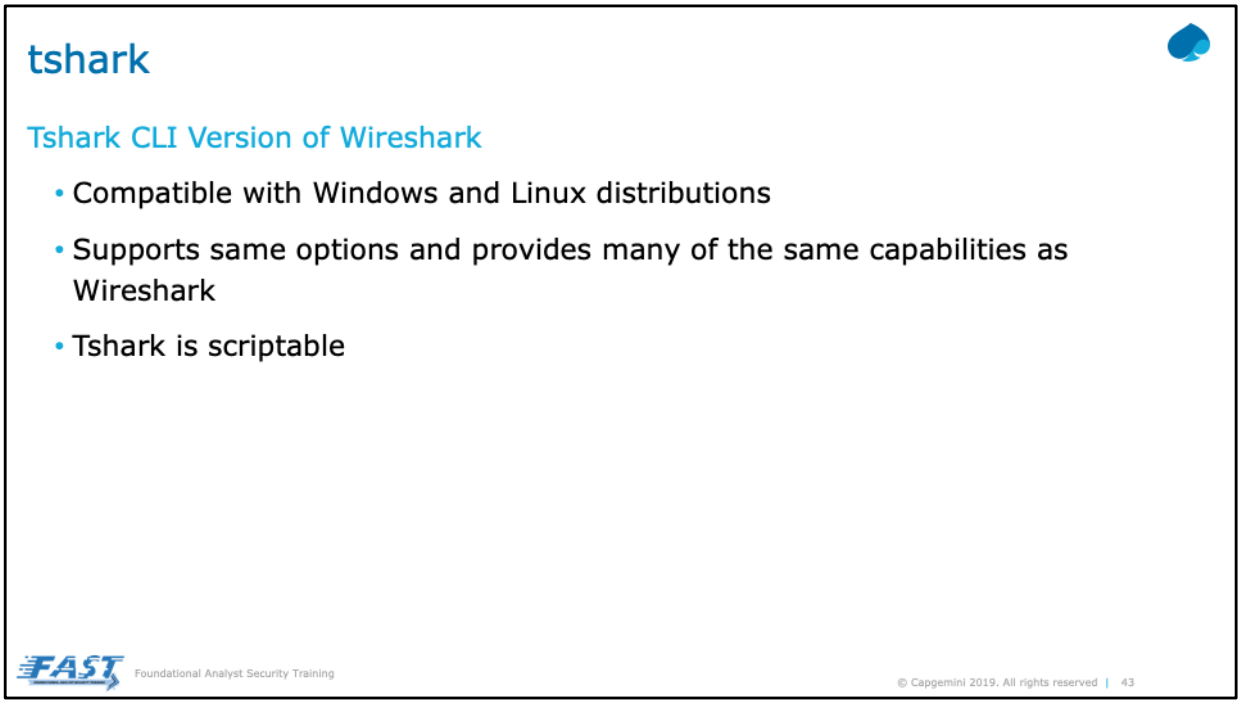
Wireshark has built-in diagnostics of protocols and traffic flows. It can identify errors, anomalous activity, and other miscellaneous activity within network traffic.

However, that Fidelity varies by protocol, but the common protocols have very robust experts and high fidelity.



Update image

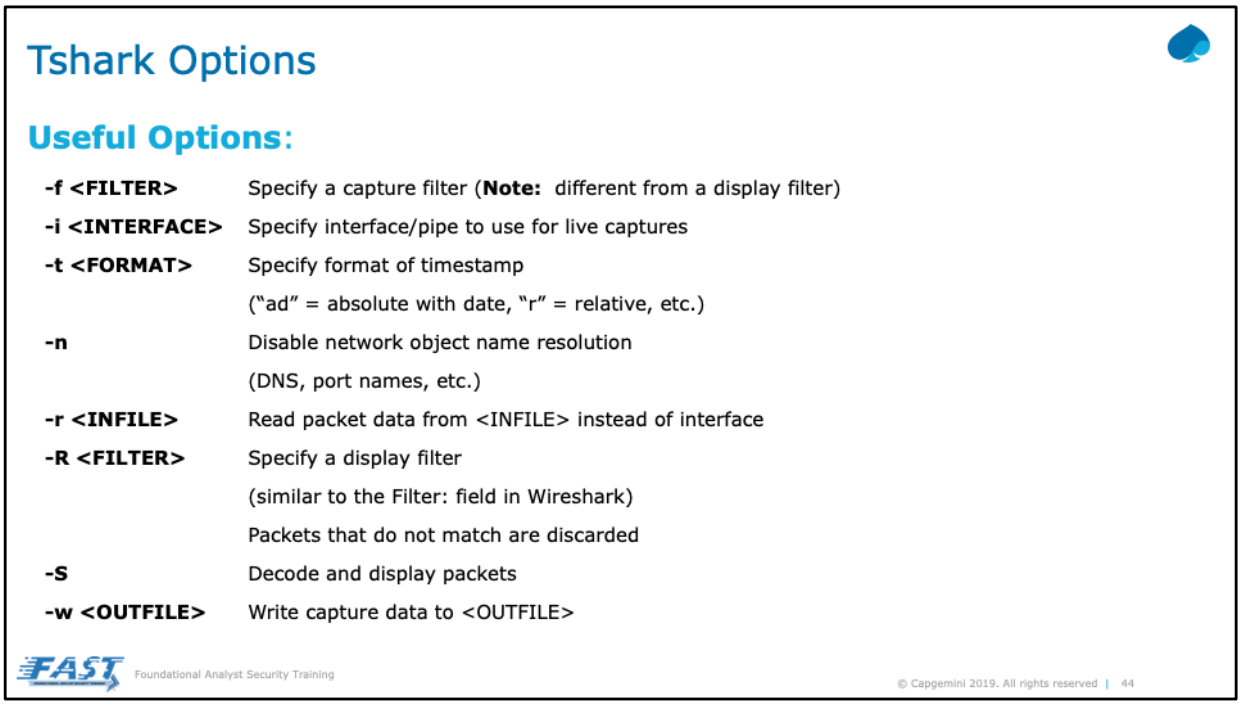
Wireshark will use a color coded display:
- Gray: information (e.g., TCP SYN flag set)
- Cyan: notable; error codes (e.g., HTTP 404)
- Yellow: warning (e.g., out of order segment)
- Red : error (e.g., malformed packet)



tshark is a command-line interface (CLI) version of Wireshark.

tshark is compatible with Windows and Linux distributions.

tshark supports same options and provides many of the same capabilities as Wireshark; however, tshark is scriptable.



Useful tshark options include:

- -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

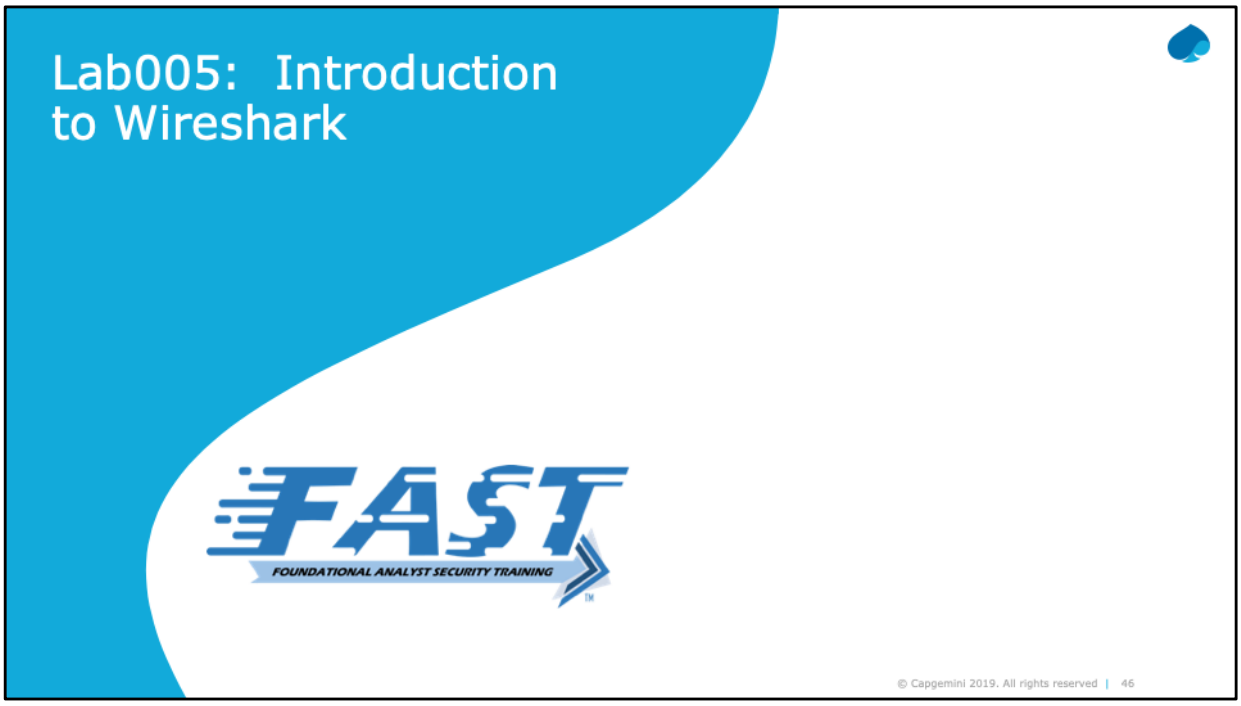
- Capture packets, and display only packets from 10.12.200.0/24.
  - # tshark –i eth2 -t ad –n –S –R "ip.addr==10.12.200.0/24"
- Read in file "web.pcap," display only HTTP traffic that contains the string ".pdf," and save the results to a new file.
  - # tshark –r web.pcap –S –R "http and frame contains '.pdf'" –w pdfs.pcap

FAST Foundational Analyst Security Training

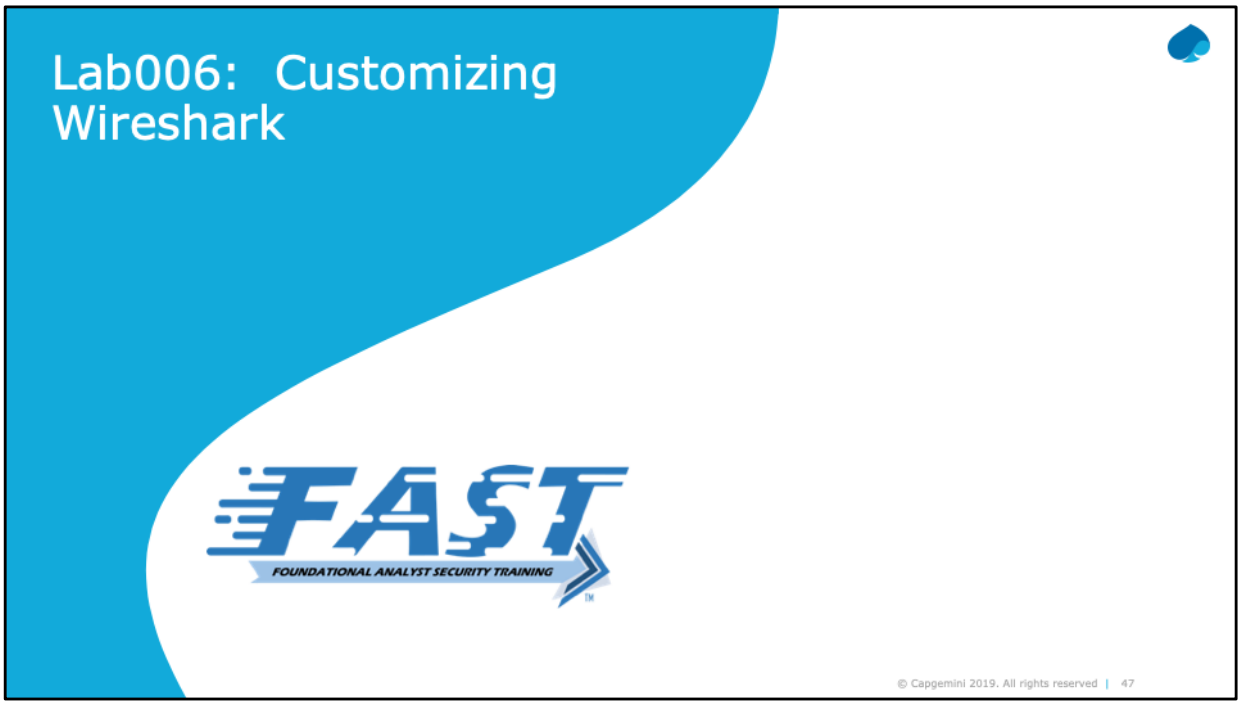
tshark examples are:

To 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"

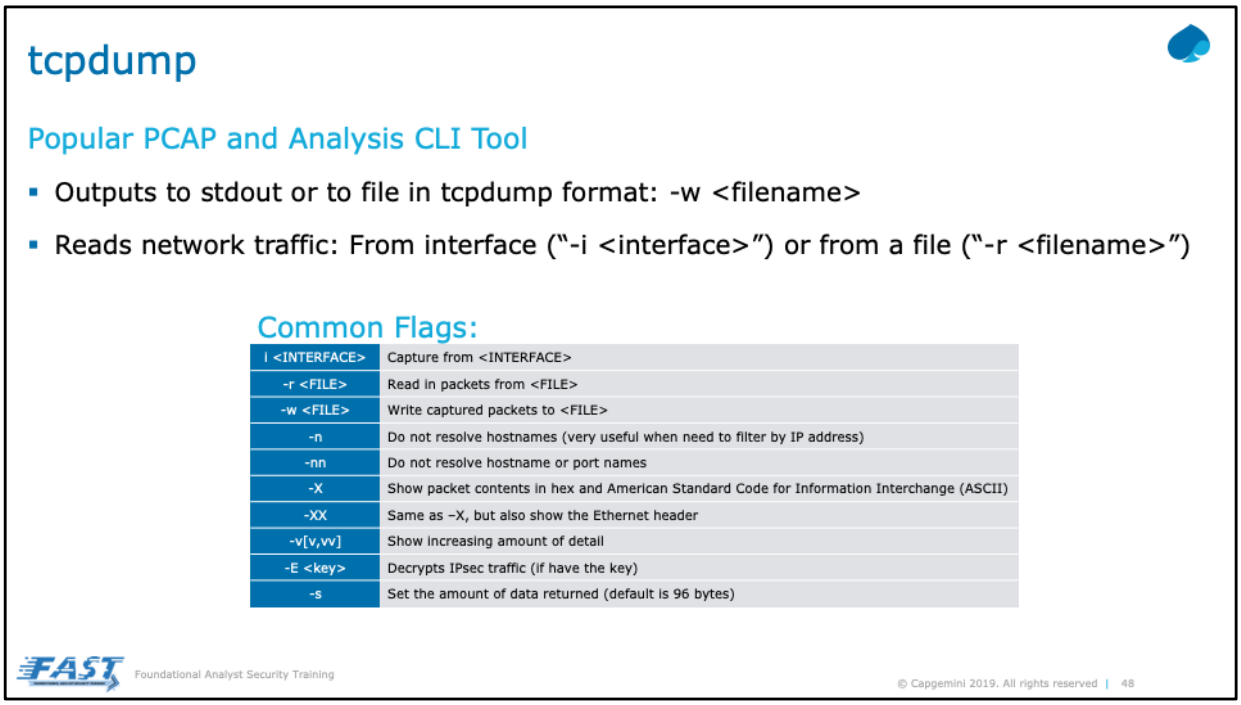
To read in file "web.pcap" and display only HTTP traffic that contains the string ".pdf" and save the results to a new file:
tshark –r web.pcap –S –R "http and frame contains '.pdf'" –w pdfs.pcap



Content Source:



Content Source:

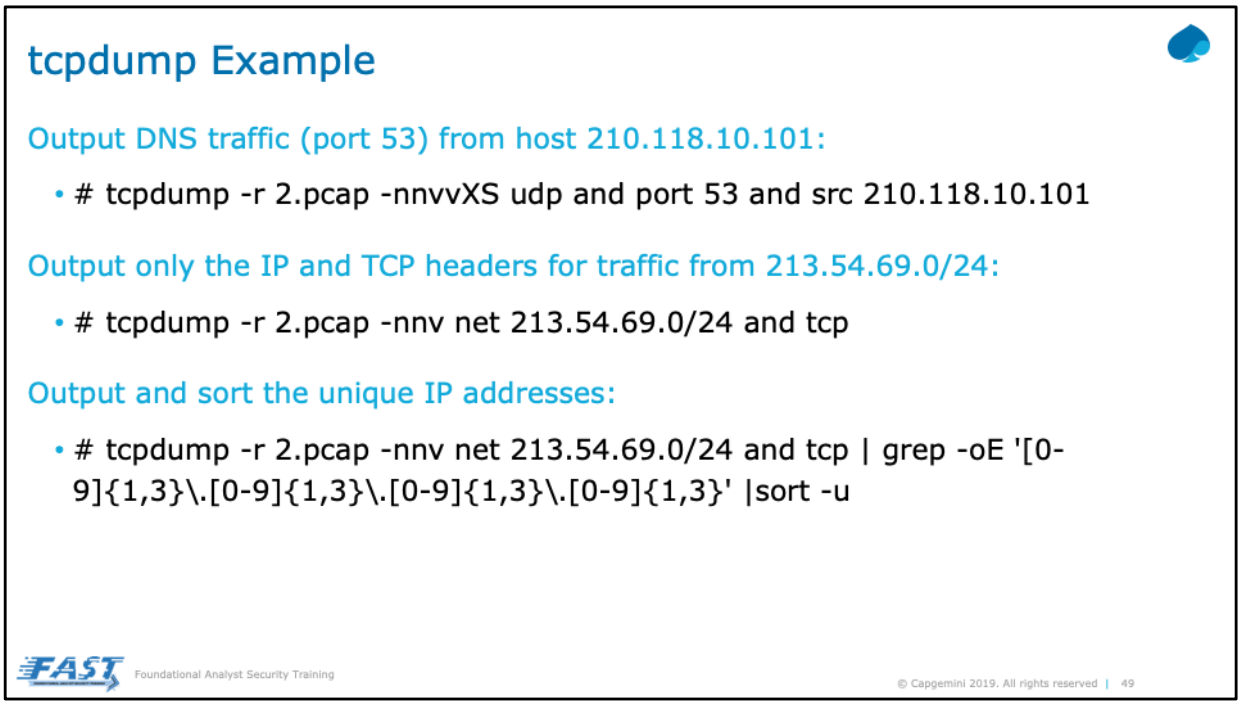


tcpdump is a popular Command Line Interface (CLI) tool used for packet capture and analysis.

You can output to stdout or to file in tcpdump format using "-w <filename>" option. You can read network traffic from interface ("-i <interface>") or from a file ("-r <filename>").

Common Flags include:

- -i <INTERFACE> Capture from <INTERFACE>
- -r <FILE> Read in packets from <FILE>
- -w <FILE> Write captured packets to <FILE>
- -n Don't resolve hostnames (very useful when you need to filter by IP address)
- -nn Don't resolve hostname or port names
- -X Show packet contents in hex and 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 you have the key)
- -s Set the amount of data returned (default is 96 bytes)

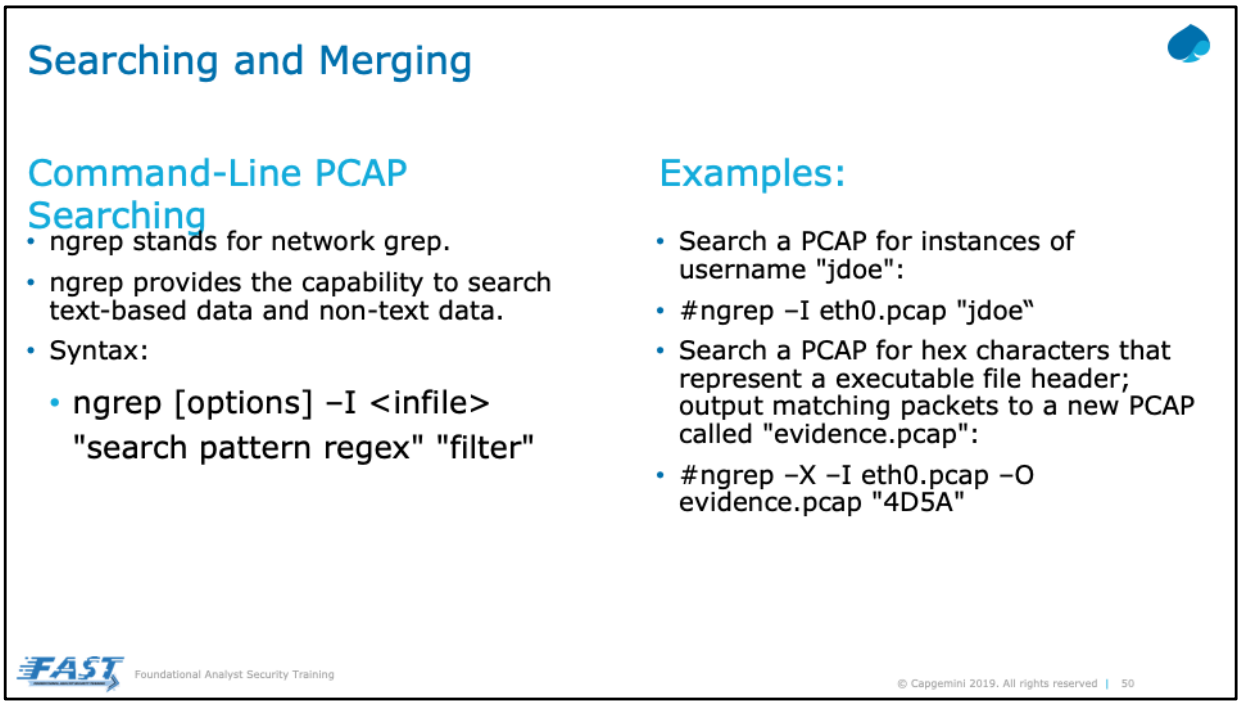


tcpdump examples are:

To 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

To 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

To 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



Ngrep enables you to do command-line PCAP Searching.

ngrep stands for network grep. It provides the capability to search text-based data and non-text data, such as images, sound files, and executables.

The syntax for an ngrep search is:
- ngrep [options] –I <infile> "search pattern regex" "filter“

ngrep examples are:

To search a PCAP for instances of username “bjones":
- #ngrep –I eth0.pcap “bjones“

To 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:

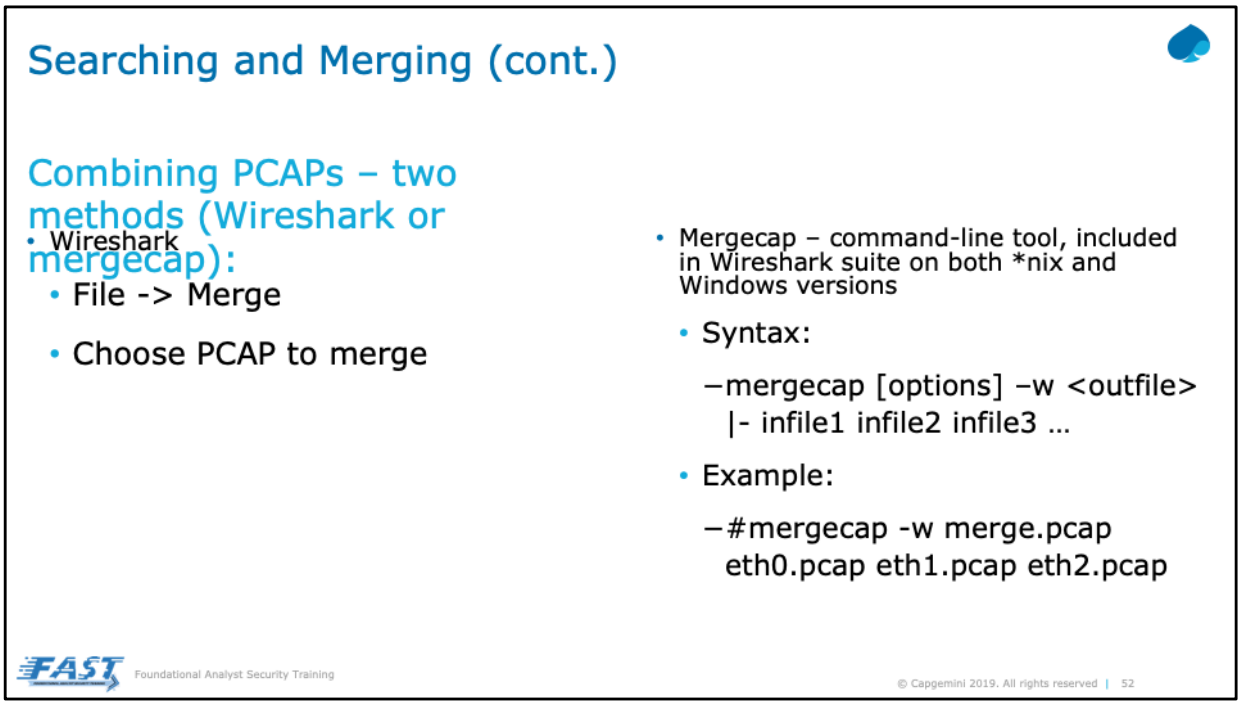
- #ngrep –I eth0.pcap "USER" "tcp and port 110"

FAST Foundational Analyst Security Training

This slide depicts a couple more ngrep examples:

To 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"

To search for clear-text POP3 logins:
- #ngrep –I eth0.pcap "USER" "tcp and port 110“



You will often need to combine PCAPs. Here are two methods you can use (Wireshark or mergecap).

In Wireshark, select File -> Merge; and then choose PCAP to merge.

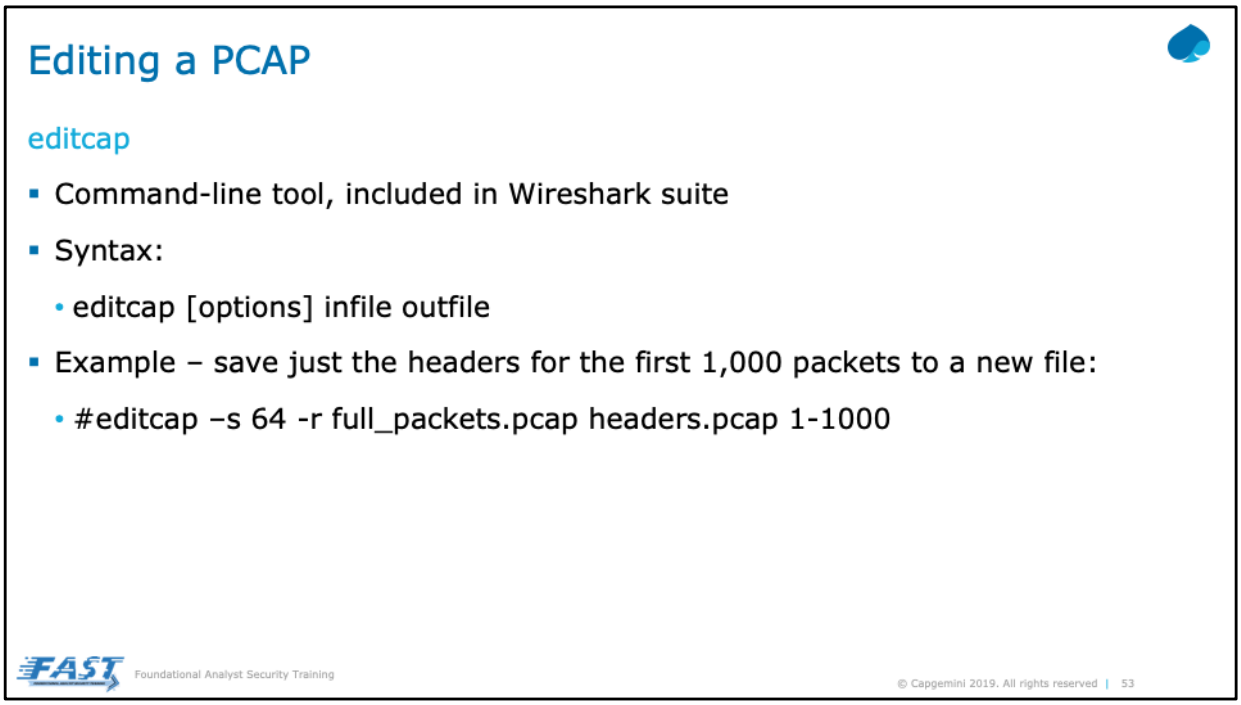
Mergecap, on the other hand, is a command-line tool, included in Wireshark suite on both *nix and Windows versions.

The syntax is:

- mergecap [options] –w <outfile> |- infile1 infile2 infile3 …

An example using mergecap is:

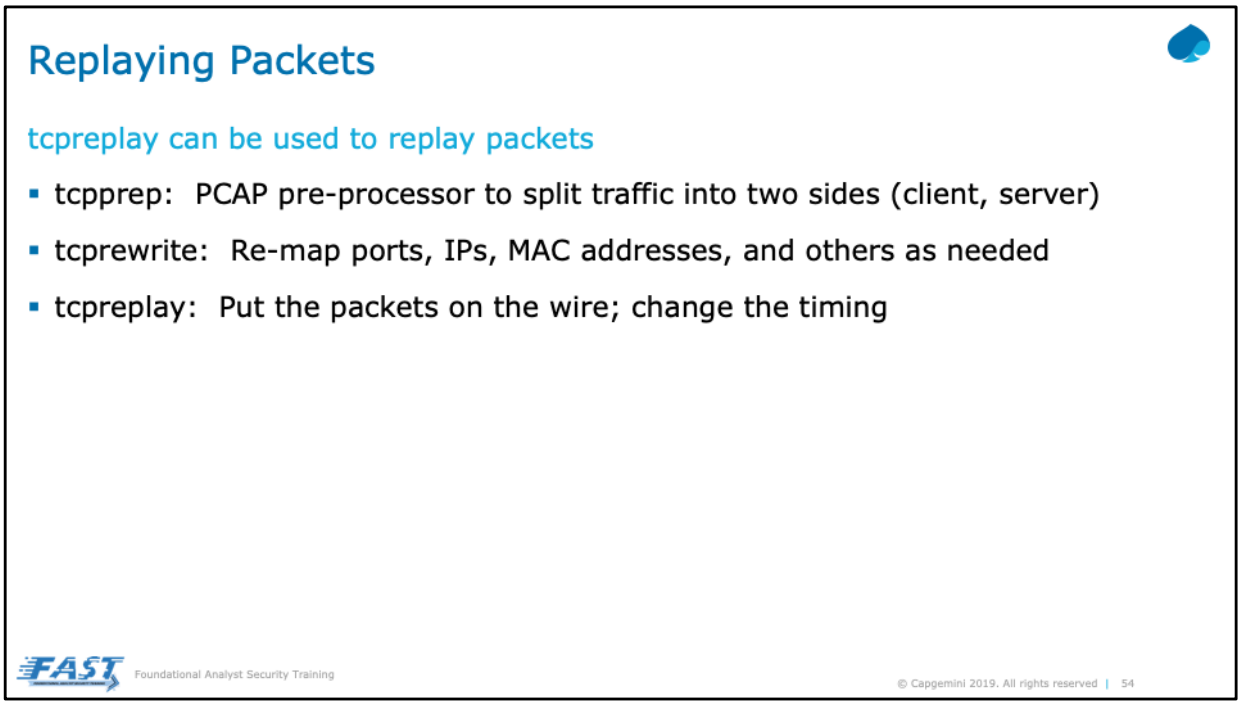
- #mergecap -w merge.pcap eth0.pcap eth1.pcap eth2.pcap



You can also edit PCAPs. This slide depicts three methods (tcpdump, tshark, editcap).

editcap is a command-line tool, included in Wireshark suite. The syntax is:
- editcap [options] infile outfile

An editcap example to save just the headers for the first 1000 packets to a new file is:
- #editcap –s 64 -r full_packets.pcap headers.pcap 1-1000



Some analysis tools can accept a packet capture file as an input for offline analysis, but you will often need to replay the packets.

tcpreplay can be used to put captured packets back on the wire. It is a suite of many tools to help prepare the packet capture.

- 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

FAST Foundational Analyst Security Training

An example to replay client/server traffic through an 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/

Foundational Analyst Security Training

Tcpflow and tcpxtract are tools to extract and reconstruct conversation.

tcpflow is a command-line tool to parse, reassemble, and extract payloads of any TCP stream it finds in a libpcap packet capture. A tcpflow to extract TCP flows from IP address 210.118.1.124 is:

- # tcpflow –v –r capturefile.pcap 'host 210.118.1.124'

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. A tcpxtract example to extract all recognizable files from a packet capture to a specific directory is:

- # 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)

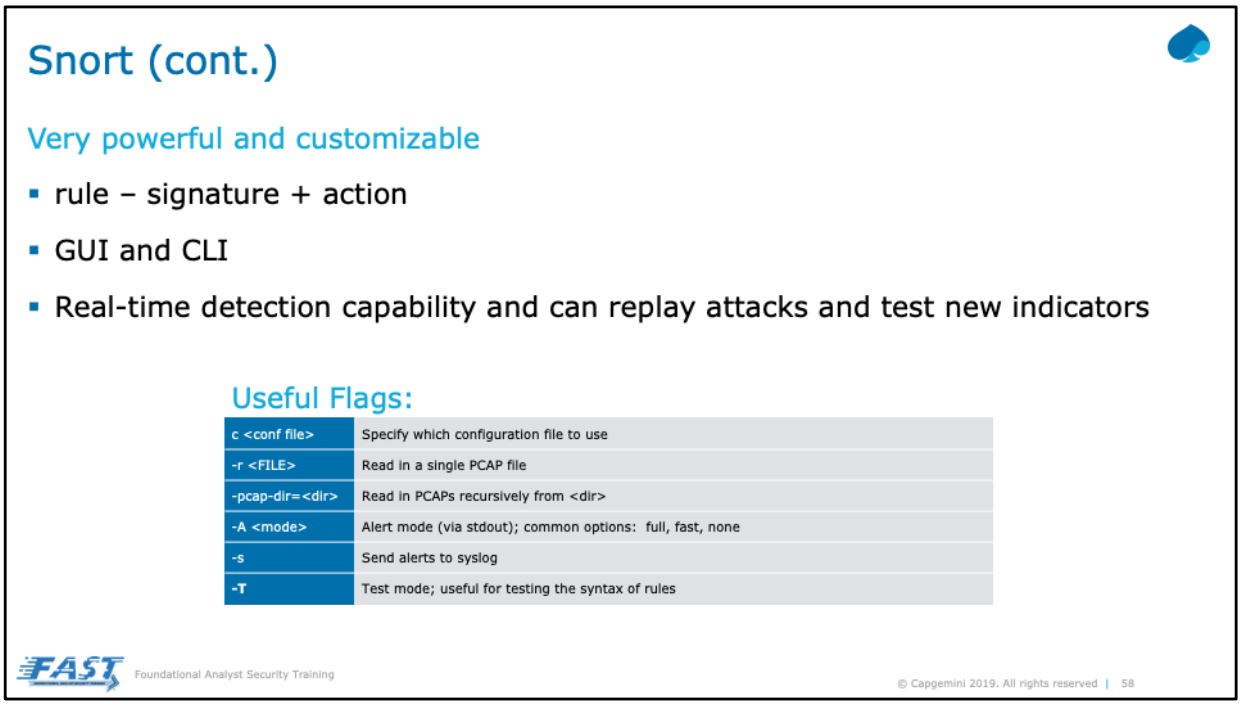
FAST Foundational Analyst Security Training

 | 57

Snort is a very common Network Intrusion Detection System (NIDS).

Snort can be configured to run in three modes:
- Sniffer mode (reads network packets and displays them)
- Packet Logger mode (logs packets to a disk)
- Network Intrusion Detection System (NIDS) mode (performs network traffic detection and analysis)

- Most IDS use or can accept snort signatures.



Snort is very powerful and customizable.

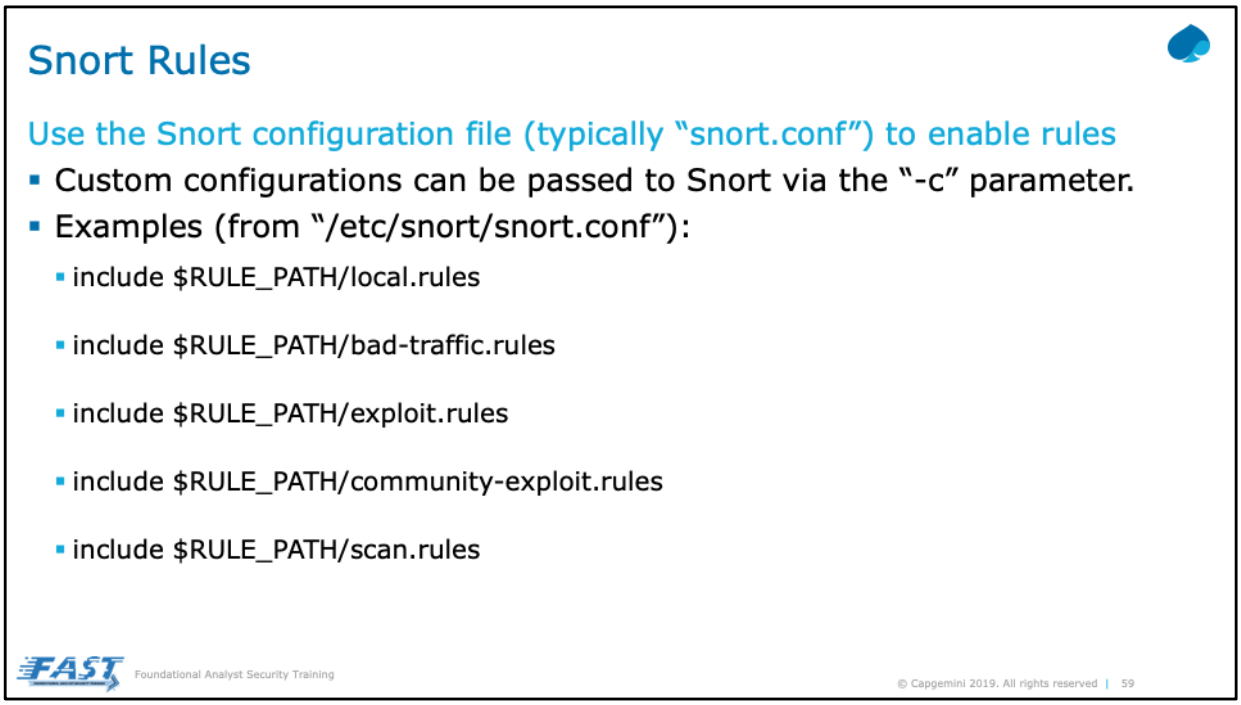
- A rule is a signature + action; a description of a network event (signature) and the action that should occur when there is a match.

- Snort was initially designed as a CLI tool, though GUI front-ends are available.

- Snort provides real-time detection, and it is useful for replaying attacks and testing new indicators.

Useful flags include:

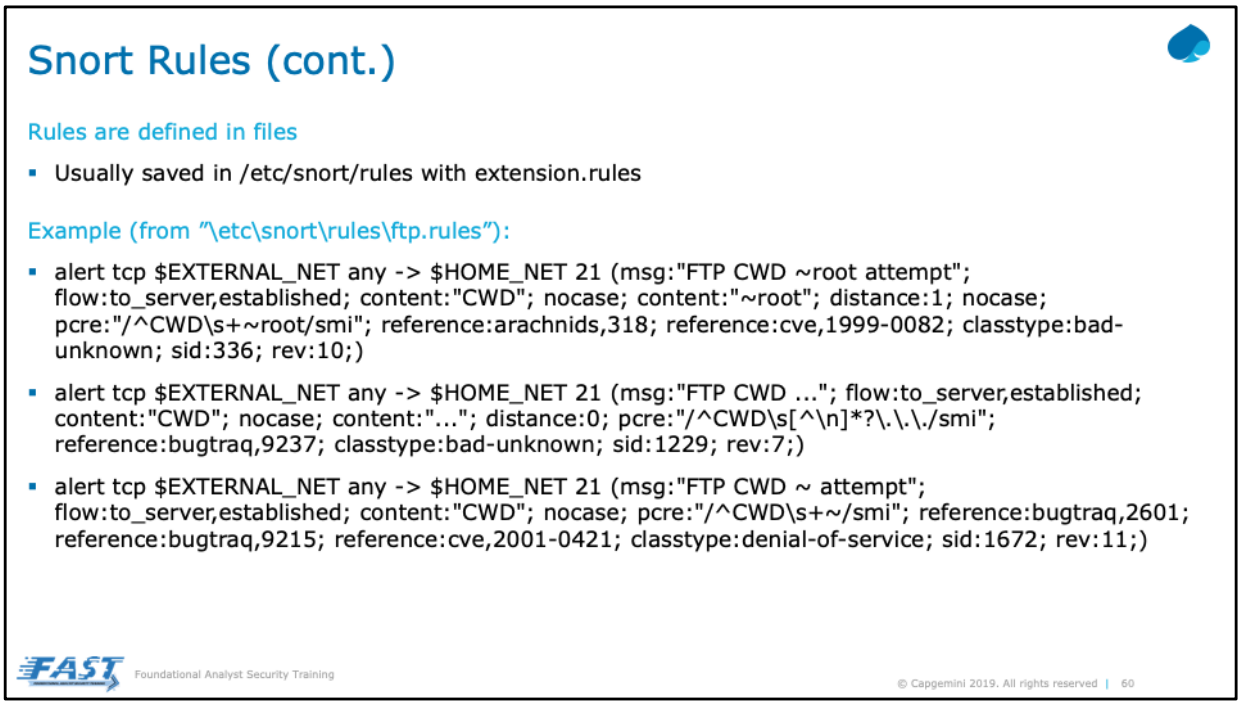
- -c <conf file> Specify which configuration file to use
- -r <FILE> Read in a single PCAP file
- -pcap-dir=<dir> Read in PCAPs recursively from <dir>
- -A <mode> Alert mode (via stdout); common options: full, fast, none
- -s Send alerts to syslog
- -T Test mode; useful for testing the syntax of your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



Rules are enabled via the Snort configuration file (typically "snort.conf"). Custom configurations can be passed to Snort via the "-c" parameter.

Example (from "/etc/snort/snort.conf") are:

- 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



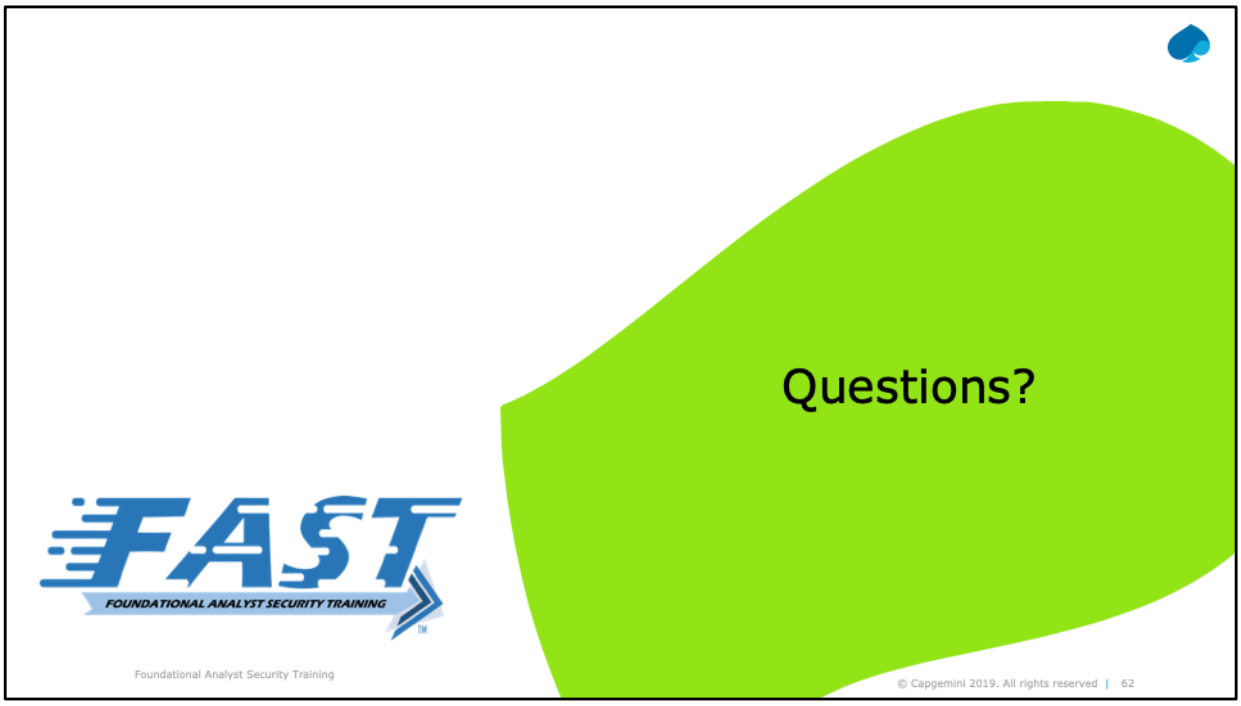
Rules are defined in files, usually saved in /etc/snort/rules with extension.rules.

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;)

A good Snort reference is: http://manual.snort.org/node26.html

Content Source:


Questions?
FAST
FOUNDATIONAL ANALYST SECURITY TRAINING
Foundational Analyst Security Training
© Capgemini 2019. All rights reserved | 62

This module will cover an in-depth look at networking technologies and topics and network forensics and analysis.

Agenda

TCP/IP AND PROTOCOLS | NETWORK DEVICES

- TCP/IP Protocols
- Network Devices
- Network Protocols & Technologies

Topic Learning Objectives

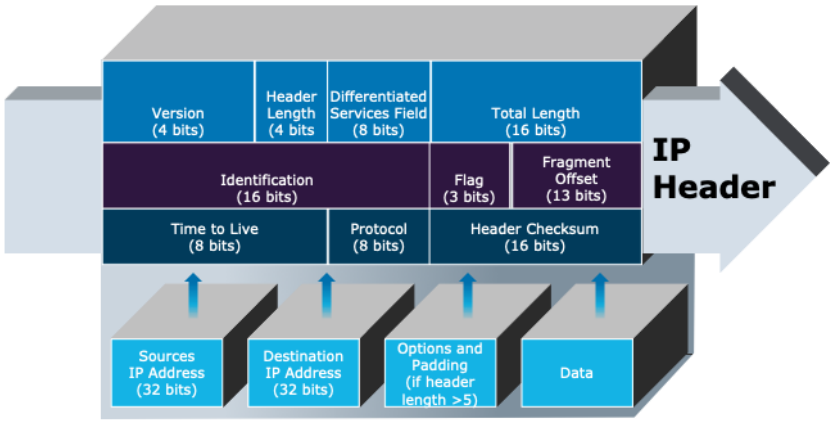
Upon completion of this topic, the student should be able to do the following:

- 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:



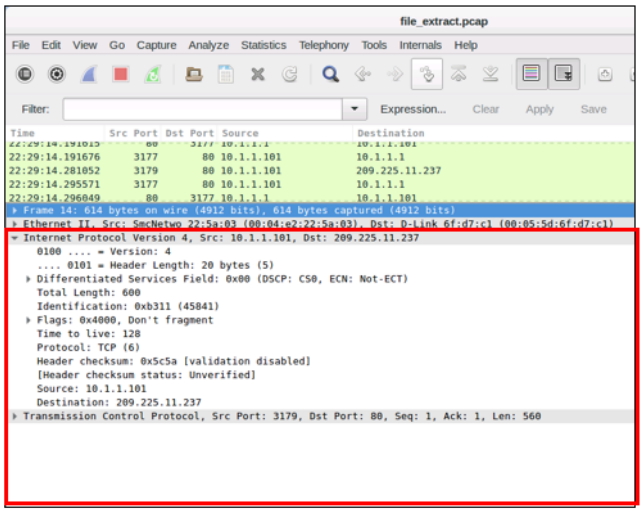
FAST Foundational Analyst Security Training

The Internet Protocol (IP) contains the fields necessary to route traffic from one network to another, including the source and destination IP addresses. Internet Protocol is a connectionless protocol.

The IPv4 Header and the fields contained within that header are depicted on this slide.

IPv4 Header – Example

IPv4 Header as seen in Wireshark:

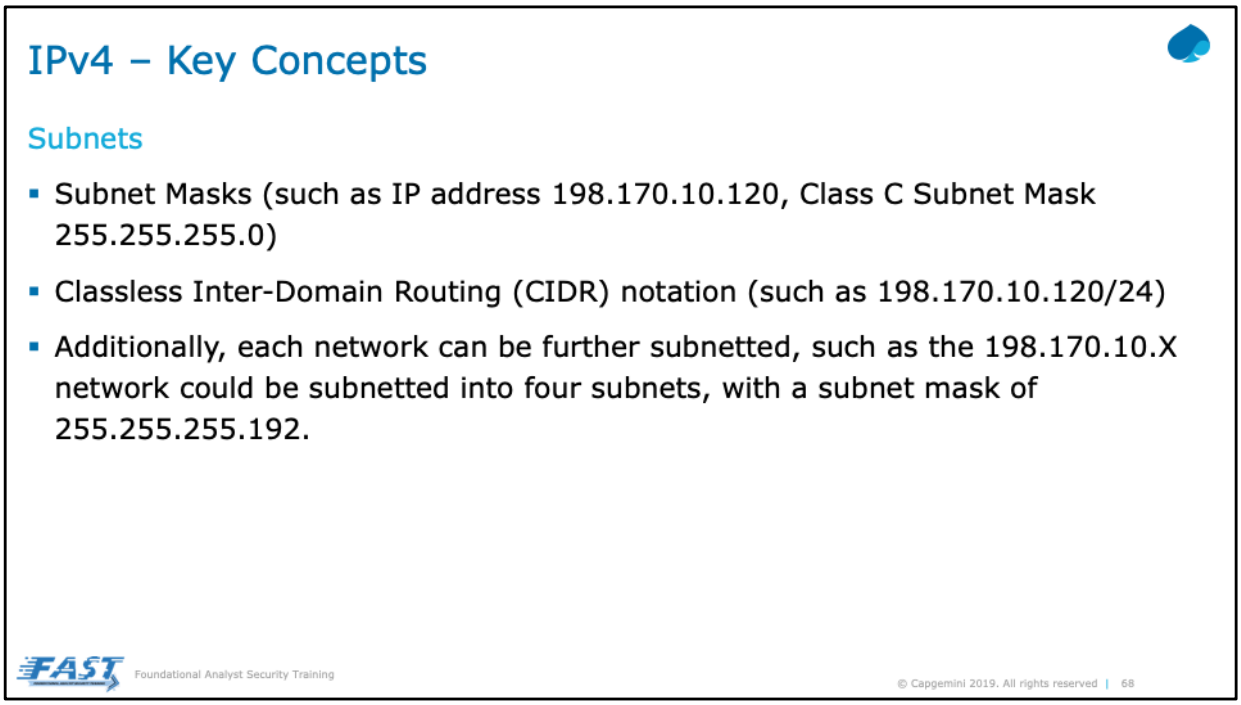


FAST Foundational Analyst Security Training

Update image

IP Header viewed in Wireshark is depicted on this slide, which shows the same basic information depicted on the previous slide but in a different layout.

At the bottom of the slide, you can see the same IP Header viewed in tcpdump.



This slide depicts address ranges and subnets.

The first bit of a Class A network begins with a 0, thus the range of the first octet can go from 0 to 127 (00000000 to 01111111), where 0 is not used and 127 is also not used, since 127 is used for loopback.

The first two bits of a Class B network begins with a 10, thus the range of the first octet can go from 128 to 191 (10000000 to 10111111).

The first three bits of a Class C network begins with a 110, thus the range of the first octet can go from 192 to 223 (11000000 to 11011111).

The first four bits of a Class C network begins with a 1110, thus the range of the first octet can go from 224 to 239 (11100000 to 11101111).

The first five bits of a Class C network begins with a 1111, thus the range of the first octet can go from 240 to 254 (11110000 to 11111111); 255 is not used.

IP address classes can be broken down into smaller segments, or subnets. In order to

do that, you have to define a subnet mask. However, each Class has a default subnet mask, such as a Class C default subnet mask is 255.255.255.0.

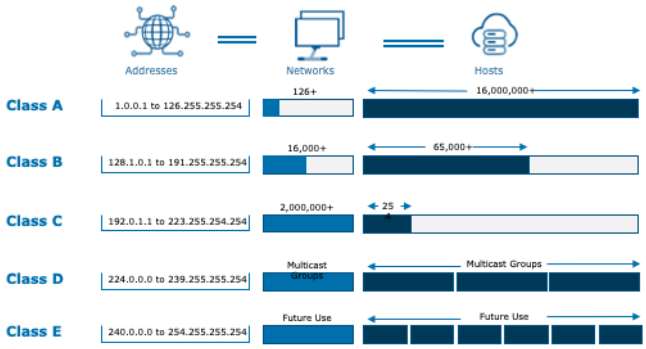
Thus, an IP address of 198.170.10.120 with a subnet mask of 255.255.255.0, would not further divide the Class C address range and there would be 265 addresses available (actually 256-2, which is 254).

You can also use Classless InterDomain 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. The 192 for the last octet, would be the subnet mask to divide the Class C address range into four subnets, because for four subnet, you would need to use two bits (128+64 = 192).

RFC1918 Name

IP Address Ranges

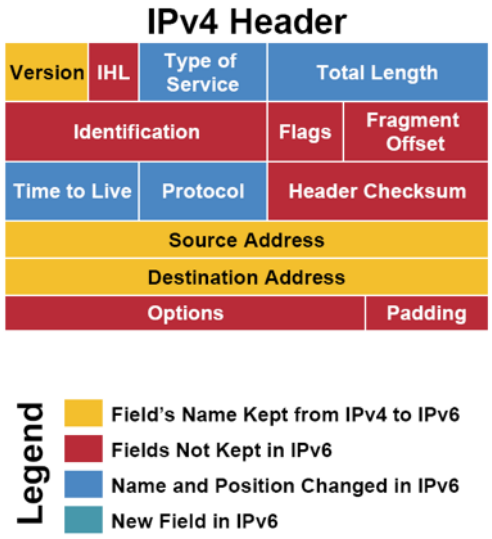


Class	Address Range	# Networks	# Hosts
A	1.0.0.1 to 126.255.255.254	126	16,000,000+
B	128.1.0.1 to 191.255.255.254	16,000+	65,000+
C	192.0.1.1 to 223.255.254.254	2,000,000+	254
D	224.0.0.0 to 239.255.255.254	Multicast Groups	Multicast Groups
E	240.0.0.0 to 254.255.255.254	Future Use	Future Use

Network Layer (Layer 3)

IPv4

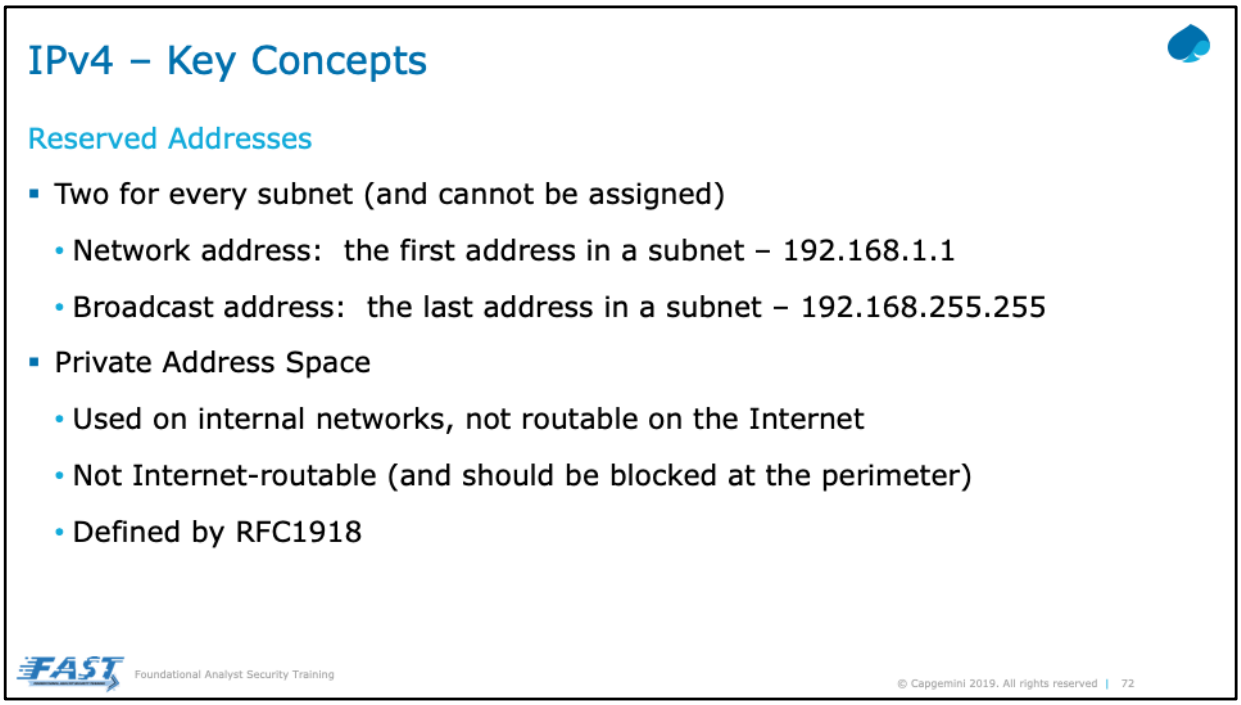
- 4 sets of decimal numbers from 0 to 255
- This is referred to as "dotted-decimal" notation.
- Examples:
  - 192.168.0.1 – Non-Routable Address
  - 127.0.0.1 – Loopback or Home Address
  - 8.8.8.8 – Common DNS Address



FAST Foundational Analyst Security Training

 | 71

IPv4 is described as 4 sets of decimal numbers from 0 to 255. This is referred to as "dotted-decimal" notation. Example IPv4 addresses would be 192.168.0.1, 127.0.0.1, and 8.8.8.8. There are several special address types: Loopback, broadcast, unicast, and multicast.



You also need to be familiar with Reserved and Private addresses.

There are two Reserved Addresses in every subnet that can not be assigned. Those are the Network address, which is the first address in a subnet, and the Broadcast address, which is the last address in a subnet.

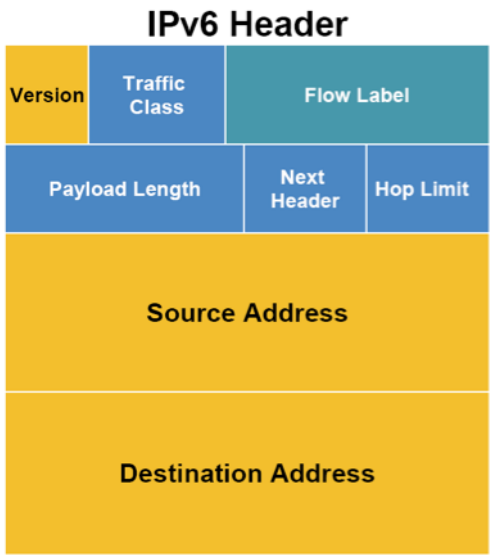
There are also Private Address ranges, which cannot be assigned as externally routed addresses. They are used on internal networks, and are not routable on the Internet. Since they are not Internet-routable, they should be blocked at the perimeter.

Private addresses are defined in RFC1918 and are depicted on this slide.

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



FAST Foundational Analyst Security Training

IPv6 addresses have 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. Leading zeros in a group can be omitted, and leading zeros can be combined once with two consecutive colons (::). An example IPv6 address is 2001:0db8:0:130F::87C:140B = 2001:0db8:0000:130F:0000:0000:087C:140B. ::1 = 0000:0000:0000:0000:0000:0000:0000:0001. More address types include loopback, multicast, link-local unicast, site-local unicast, and global unicast.

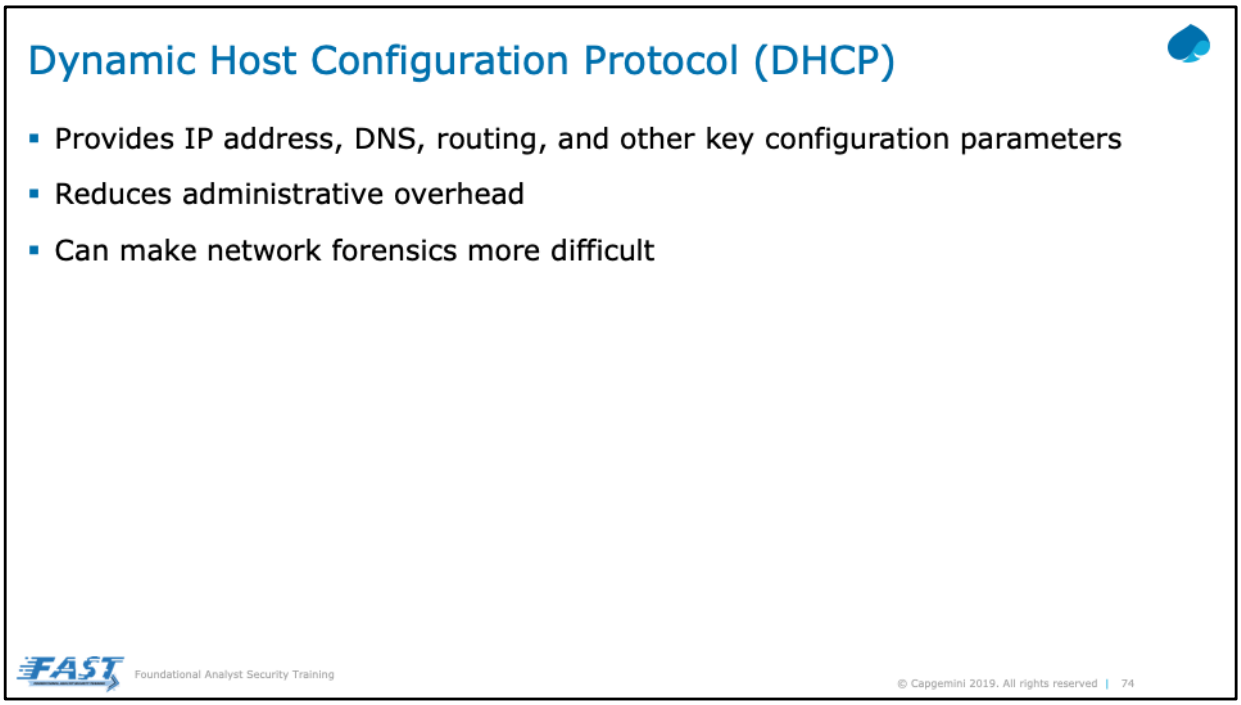
IPv6 uses 128-bit addressing (versus 32-bit addressing in IPv4).

IPv6 uses eight groups of four hexadecimal digits, separated by colons. Each hexadecimal digit represents 4 bits.

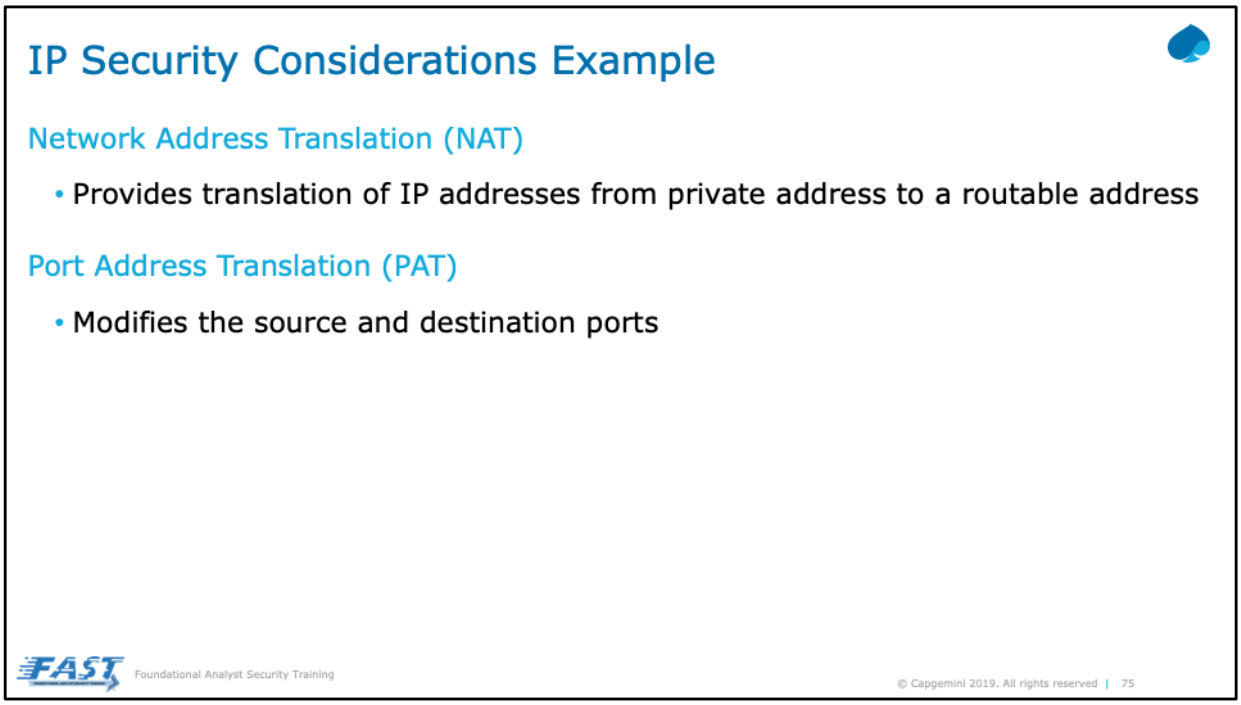
- 36bd:18ac:4545:38da:200f:f5ff:fe33:96cf

Beyond the significant increase in number of bits, IPv6 has various benefits, including:

- It handle packets more efficiently.
- It has improved performance.
- It has increased security features.



Dynamic Host Configuration Protocol (DHCP) is a protocol that provides IP address, DNS, routing, and other key configuration parameters dynamically. A major advantage of DHCP is that it significantly reduces administrative overhead. However, it can make network forensics more difficult.

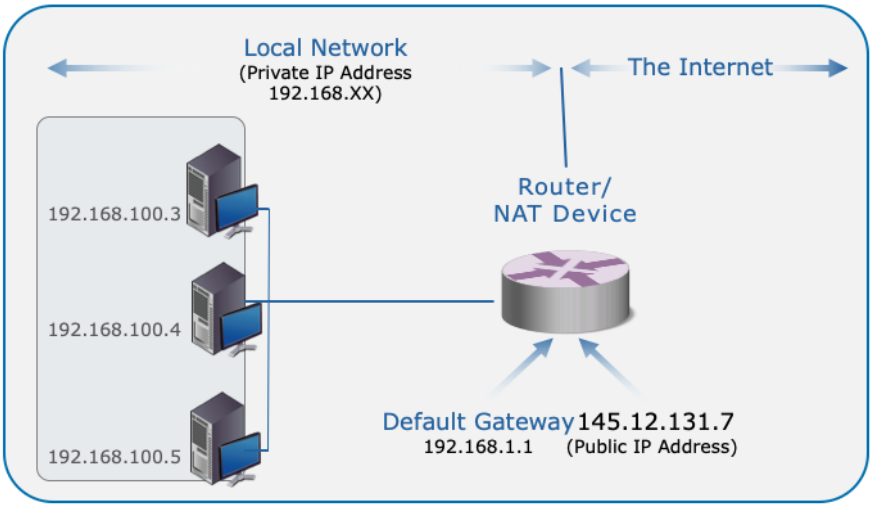


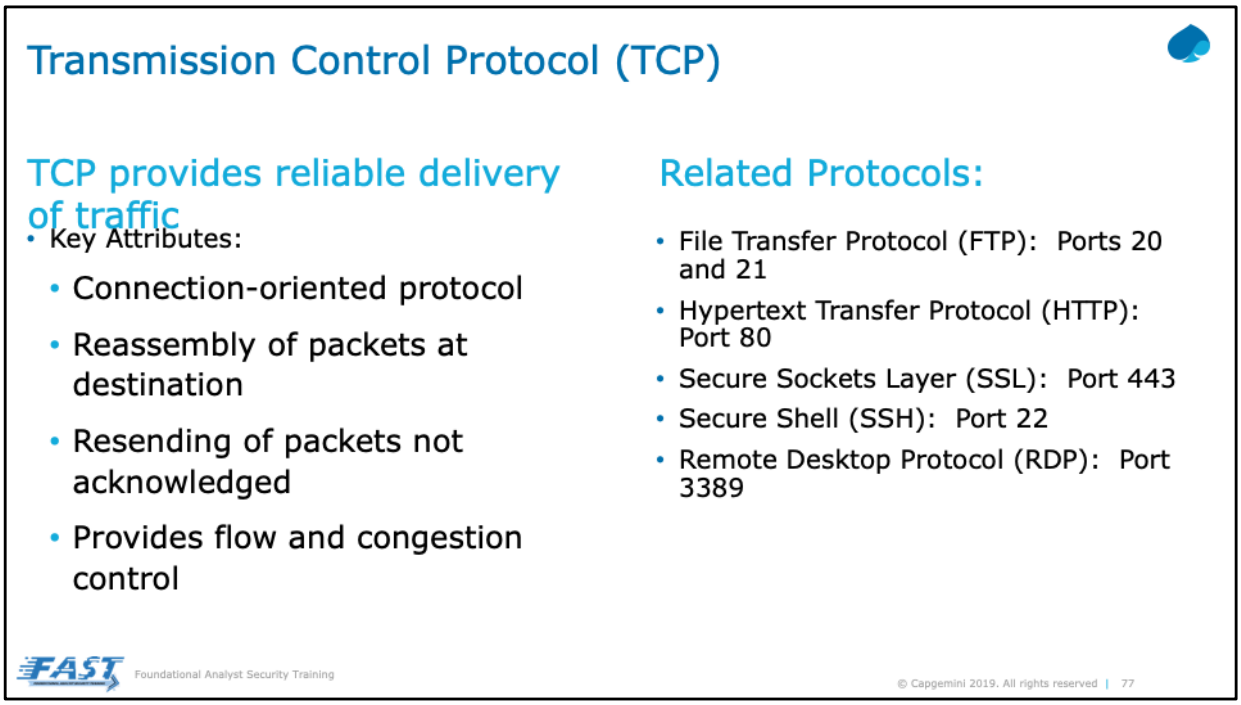
Network Address Translation and Port Address Translation were designed for security and conservation purposes.

Network Address Translation (NAT) provides translation of IP addresses from Private address to a routable address.

Port Address Translation (PAT) provides translation of the source and destination ports.

IP Routing





TCP is connection-oriented and provides reliable delivery of traffic.

Key attributes of TCP include:
- It is a connection-oriented protocol.
- It reassembles packets at the destination.
- It resends packets not acknowledged.
- It provides flow and congestion control.

Other related Protocols include:
- File Transfer Protocol (FTP): Ports 20 & 21
- Hypertext Transfer Protocol (HTTP): Port 80
- Secure Socket Layer (SSL): Port 443
- Secure Shell (SSH): Port 22
- Remote Desktop Protocol (RDP): Port 3389

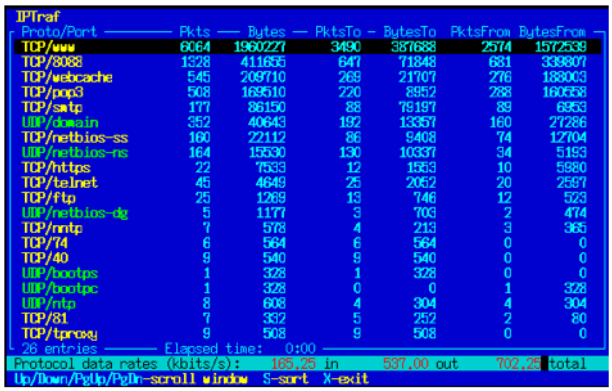
Ports

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)

FAST Foundational Analyst Security Training

Ports are used by TCP and UDP (and others) to map network services to host processes.

A Process + a Port = a Socket.

Port values can be up to a16-bit value, which is up from 1-65535.

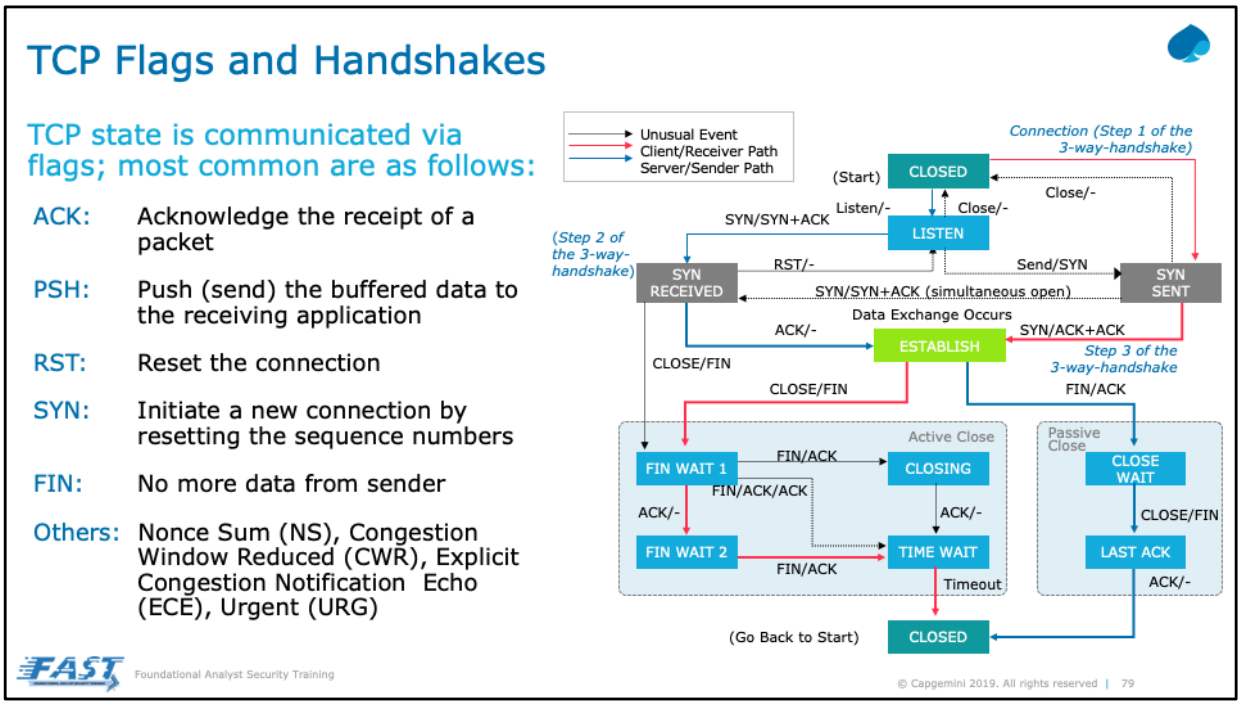
There are well-known, registered, and dynamic ports.

Well-known ports are from 0–1023, and are used by system processes to provide common network services. It usually require root privileges to bind on the host.

Registered ports were formally assigned by the IANA for use with a certain protocols or applications. They can be port numbers from 1024–49151.

Dynamic ports are those equal to or greater than 49152.

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



TCP state is communicated via flags; most common are:

- 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: NS, CWR, ECE, URG

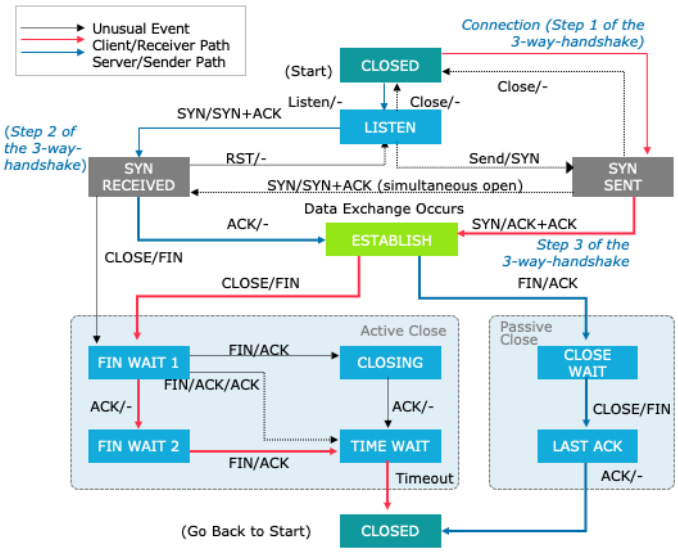
Normal Connection Establishment (the 3-way handshake):

- First, the client sends TCP packet with following the SYN flag set, and it sets sequence number to value X.
- Then, the server responds with the SYN and ACK flags set, it sets acknowledgement number to X+1, and it sets new sequence number to value Y.
- Then, the client responds with the ACK flag set, the acknowledgement number is set to X+1, and the sequence number is set to Y+1.
- Then, the connection is established.

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.



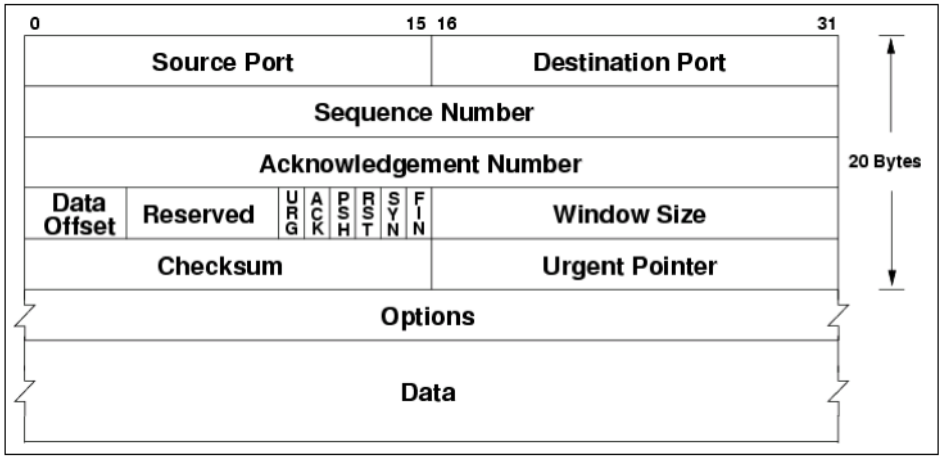
TCP state is communicated via flags; most common are:

- 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: NS, CWR, ECE, URG

Normal Connection Establishment (the 3-way handshake):

- First, the client sends TCP packet with following the SYN flag set, and it sets sequence number to value X.
- Then, the server responds with the SYN and ACK flags set, it sets acknowledgement number to X+1, and it sets new sequence number to value Y.
- Then, the client responds with the ACK flag set, the acknowledgement number is set to X+1, and the sequence number is set to Y+1.
- Then, the connection is established.

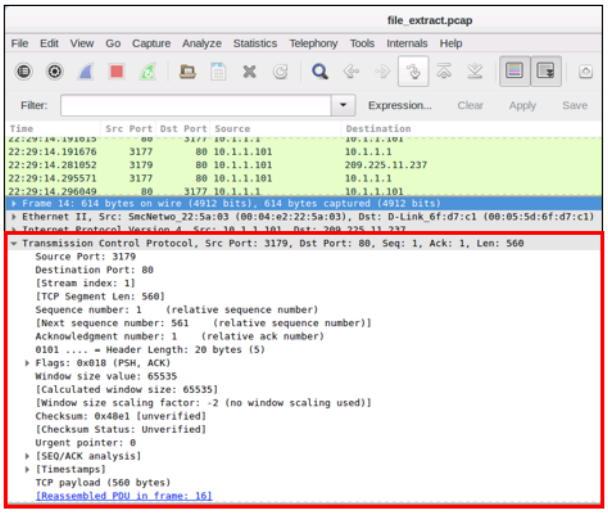
TCP Header



This slide depicts the fields and the field sizes for the TCP Header.

TCP Header Example

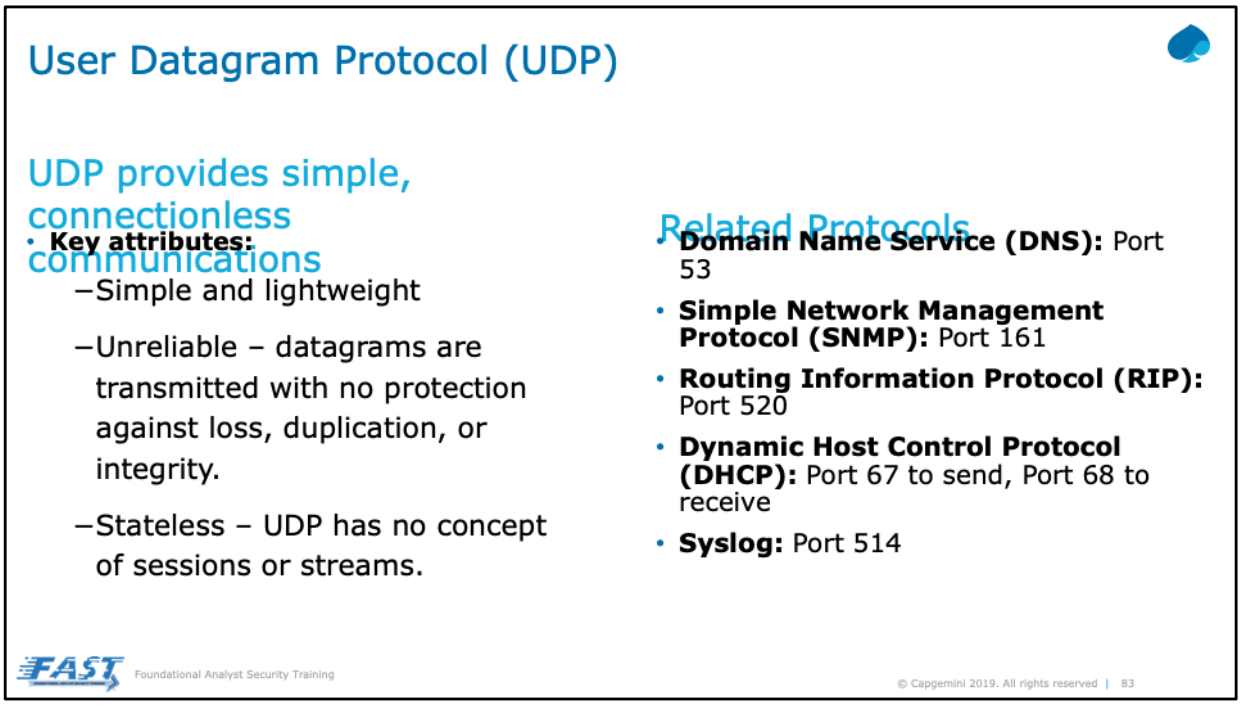
Single TCP Header Viewed in Wireshark:



FAST Foundational Analyst Security Training

Change images

This slide depicts a TCP Header in Wireshark and tcpdump.



UDP provides simple, connectionless communications.

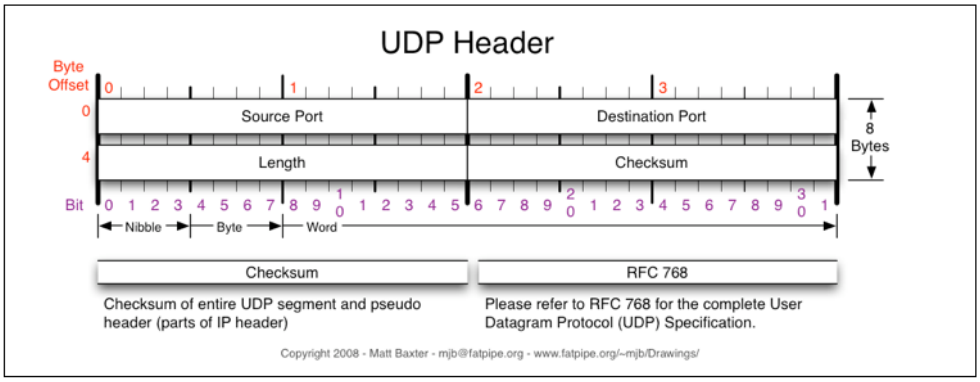
Key attributes of UDP are:

- 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 include:

- 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



Change Wireshark image

This slide depicts the fields and field sizes of a UDP Header and a sample UDP Header viewed in Wireshark.

User Datagram Protocol (UDP)

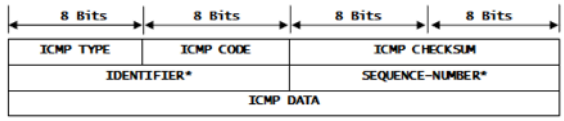
TCP	UDP
Reliable	Unreliable
Connection-oriented	Connectionless
Segment retransmission and flow control through windowing	No windowing or retransmission
Segment sequencing	No sequencing
Acknowledge segments	No acknowledgement

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

Frame format – ICMP



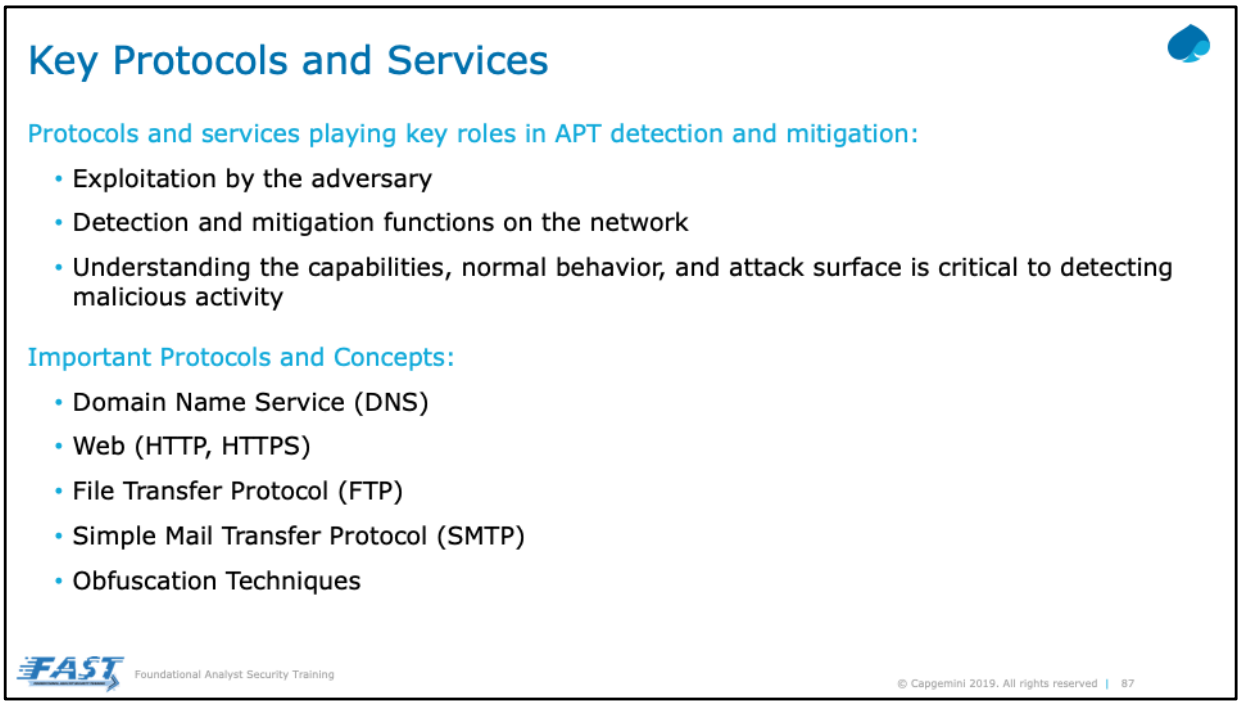
*Not part of ICMP – Can be used arbitrarily by the user

FAST Foundational Analyst Security Training

The Internet Control Message Protocol (ICMP) is designed to communicate error and diagnostic messages.

Following are the ICMP "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



There are certain network protocols and services that play key roles in the detection and mitigation of APT, either because:

- 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

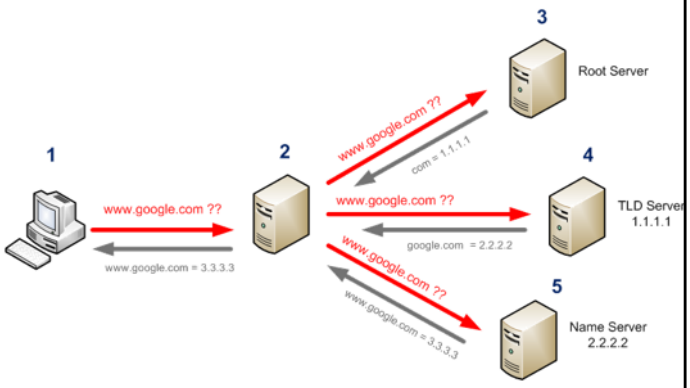
Important Protocols and Concepts include:

- 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



FAST Foundational Analyst Security Training

The early Internet landscape was pretty barren with only a few hundred computers making up the ARPANET, the military/educational precursor to the Internet. Then, as today, each device on the network was a node, and each node needed a unique address to enable data packets to find their destinations. Anyone that has ever used IP addresses knows that it's tough enough to remember a few addresses on your local network, much less keep track of the addresses for remote systems you use often. That's where host names came into the picture.

How DNS works

DNS essentially functions as a distributed database using a client/server relationship between clients that need name resolution (mapping host names to IP addresses) and the servers that maintain the DNS data. This distributed database structure enables the DNS name space to be both dynamic and decentralized, giving local domains control over their own portion of the DNS database while still enabling any client to access any part of the database.

At the uppermost level of the DNS name space are the root servers. The root servers manage the top level domains: .com, .net, .org, .mil, .edu, .gov, and .int. With all the domains in existence today, not to mention all the hosts in those domains, you can

see why the root servers actually maintain very little information about each domain. In fact, the only data the root servers typically maintain about a domain are the name servers that are authoritative for the domain, or which have authority for the domain's records.

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)

FAST Foundational Analyst Security Training

Host names provide a more "friendly" way to name hosts, making it easier to remember host addresses. For example, when you want to get the news, you can point your browser to www.newsmax.com instead of 64.29.200.227. Add a couple of hundred other addresses to your frequent site list, and you can see that host names are a lot easier on the brain than IP addresses.

But converting host names to IP addresses doesn't happen by magic. It needs some form of translation to make it happen, and the mechanism that enables that translation is the Domain Name System, or DNS.

It is important to understand DNS for many reasons. For example, unless the adversary hard-codes the Delivery and C2 IP addresses in the malcode, infected hosts must use DNS to communicate with external systems (especially when using dynamic DNS). HTTP is a common delivery method for APT, and almost all web requests are preceded by DNS lookups. DNS can also be used for command and control. Additionally, DNS provides very effective detection and mitigation capabilities, such as DNS "black holes", once malicious domain and IP addresses are known.

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

Until recently, an organization known as InterNIC was responsible for managing the majority of the top-level domains in the DNS name space. InterNIC switched from being a nonprofit organization to the now for-profit Network Solutions. When it made that switch, it lost its monopoly on the name space and now there are several entities that register and maintain the DNS name space.

There are other domain suffixes as well, including national domains such as the us domain, which is used for governmental, regional, and educational entities in the United States. Other countries have their own domains, such as jp for Japan, uk for the United Kingdom, and so forth.

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

FAST Foundational Analyst Security Training

In the early days when there were only a few hundred nodes, a single text file could easily map host names to their corresponding IP addresses. This text file, called Hosts.txt, was managed by the Standford Research Institute (SRI) and contained all of the name-to-address mappings for all nodes on the ARPANET.

Operating systems (primarily UNIX at that point) use the Hosts.txt file to map host names to IP addresses. System administrators copied the Hosts.txt file from SRI to their local systems periodically to update their address maps.

As the number of nodes on the network continued to increase, using a relatively static text file to provide mapping quickly became impractical. New hosts were added so rapidly that neither SRI nor system administrators had any hope of keeping up. So, the DNS system was developed in the mid-1980s to provide a dynamic means of updating and resolving host names to their IP addresses.

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.

FAST Foundational Analyst Security Training

Dynamic DNS (DynDNS) is a service that allows a network device to change its IP address, while maintaining the same domain name and is updated in real time. Whereas typical DNS relies on A or AAAA records that statically map IP addresses to domain names, DynDNS typically uses a client program that constantly updates the DynDNS service with the updated IP address of the server.

DynDNS accounts and subdomains can be procured quickly, freely, and anonymously. They are also easy to manage.

Botnet masters and APT adversaries regularly use DynDNS domains to allude IP address blocks and other detections by constantly changing IP addresses, while maintaining connectivity via a static domain name.

Given the number of DynDNS providers and the myriad of domains registered to these providers, DynDNS is difficult to protect against.

A blackhole is an address or range of addresses, that are unreachable either too or from the network.

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

FAST Foundational Analyst Security Training

 | 93

The Web is such a big issue for many reasons. Web traffic provides a large attack surface, due to how much it is used by users, and the Web is a common delivery method for APTs.

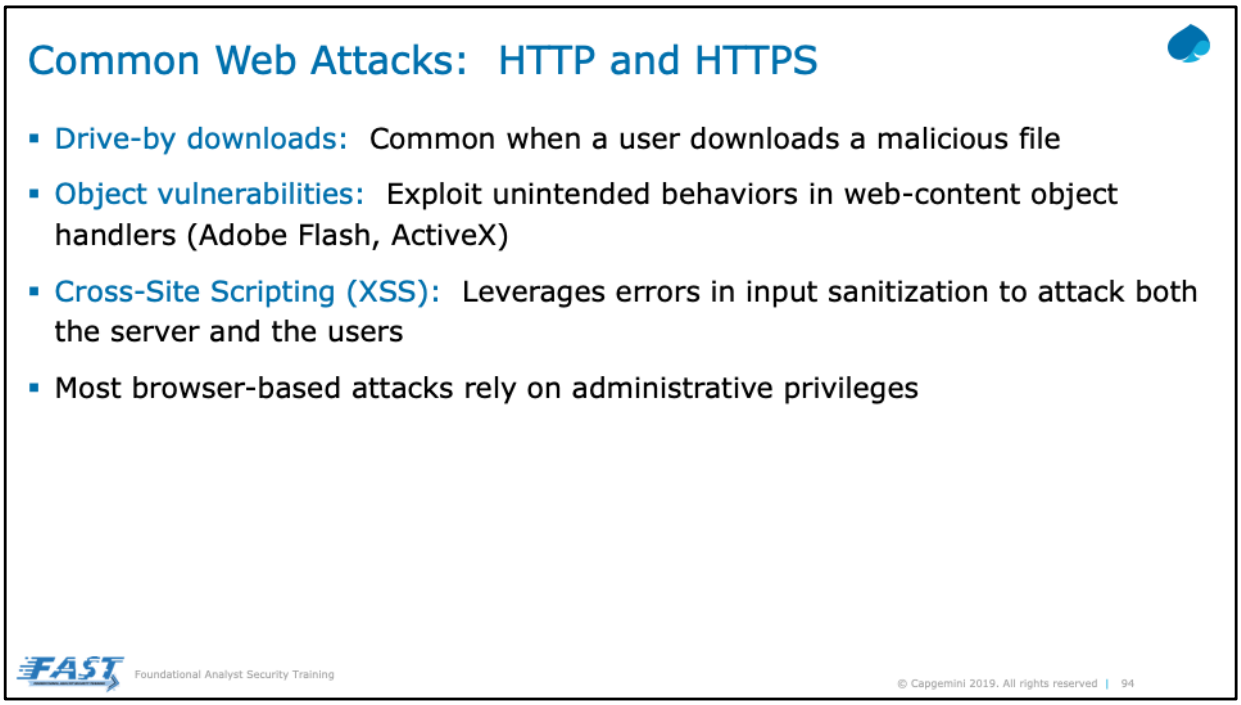
Attackers will often use “drive-by download“ attacks, and the Web is regularly used for C2 (especially HTTPS).

Port & Protocol include:

- Protocol: TCP and many others
- Port: 80 for HTTP, 443 for HTTPS

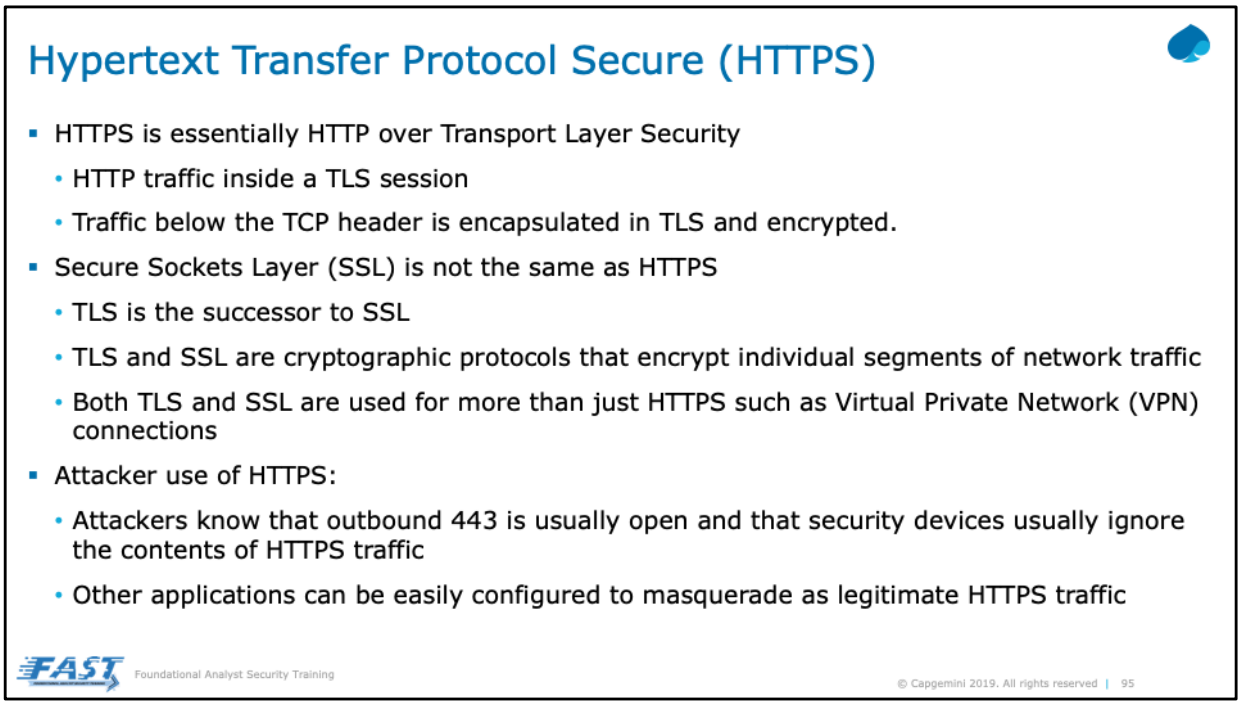
Key Points and Security Considerations are:

- Anything that understands text streams and network sockets can communicate via HTTP (not just browsers).
- Know your browser-based attacks and understand how they work (many web exploits take advantage of misconfigurations, weaknesses, or vulnerabilities).



Common Web Attacks: HTTP & HTTPS

- Drive-by downloads: Common when a user downloads a malicious file without their knowledge. Commonly accomplished via IFRAMEs and malicious scripts.
- Object vulnerabilities: Attacks that exploit unintended behaviors in web-content object handlers (e.g., Adobe Flash, ActiveX).
- Cross-site scripting (XSS): Commonly misunderstood web attack. Leverages errors in input sanitization to attack both the server itself AND THE USERS. Modern attack packages can provide full system control once a user falls victim to an XSS attack.
- Most (but certainly not all) browser-based attacks rely on the user having administrative privileges on the local system.

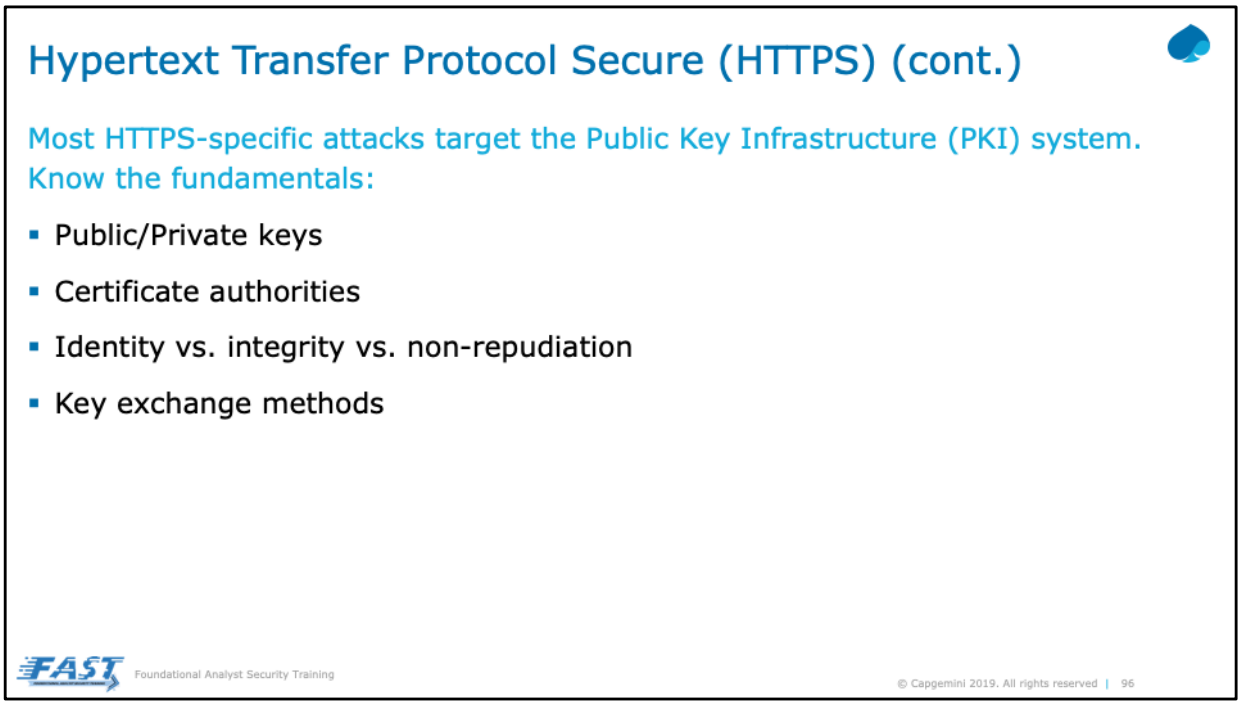


HTTPS is essentially HTTP over Transport Layer Security where HTTP traffic is inside a TLS session. All traffic below the TCP header is encapsulated in TLS and encrypted.

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 VPN connections.

Ways that an attacker uses HTTPS are:

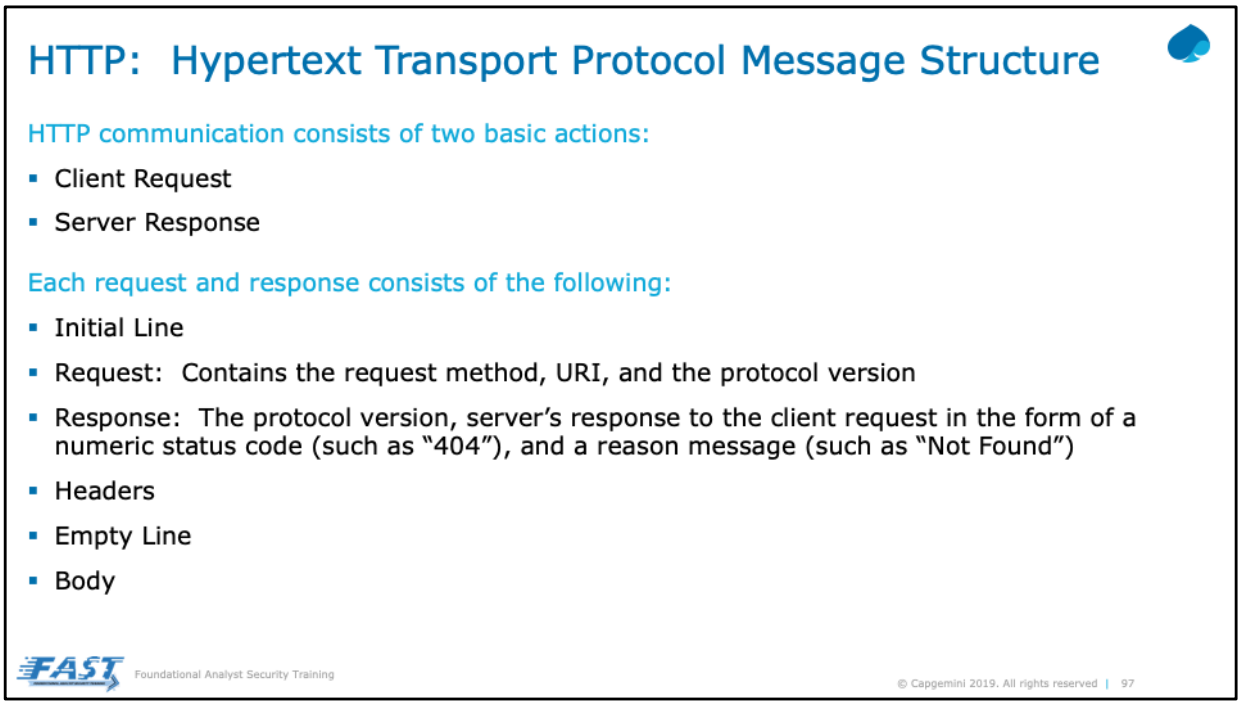
- Attackers know that outbound 443 is usually open and that security devices usually ignore the contents of HTTPS traffic.
- Other apps can be easily configured to masquerade as legitimate HTTPS traffic (such as OpenVPN tunnels over TCP/443).



Most HTTPS-specific attacks target the PKI system. Know the fundamentals of PKI:

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

These concepts will be covered in more detail in the next section.

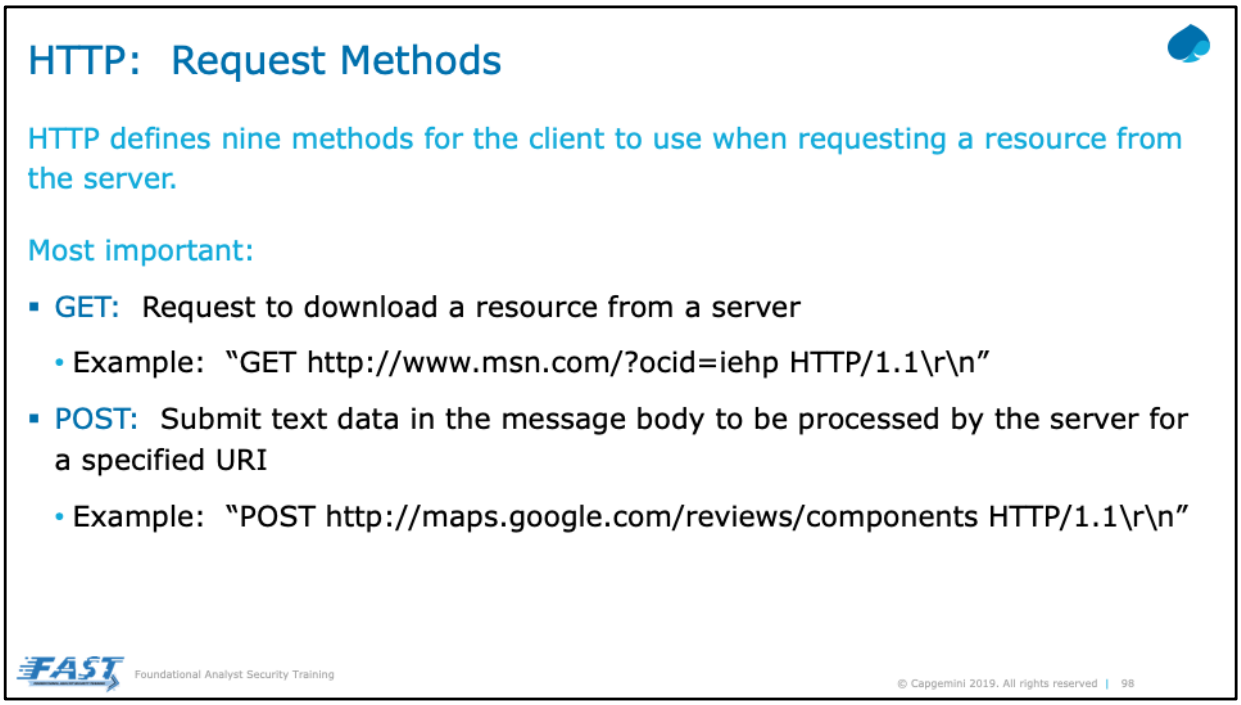


HTTP communication consists of two basic actions:

- Client Request: Client communicates with server within a TCP session indicating a request either to download a resource or upload data
- Server Response: Server responds to client request by either permitting or denying the request

Each request and response consists of:

- Initial Line
- Request: Contains the request Method, URI, and the protocol version (almost always HTTP/1.1)
- Response: The protocol version, server's response to the client request in the form of a numeric status code (e.g., "404") and a reason message (e.g., "Not Found")
- Headers
- Empty Line (to mark the end of the Headers section)
- Body (the data)



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

The most important are:

- 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

FAST Foundational Analyst Security Training

HTTP defines many status codes and messages.

The Server can also define custom codes and messages. Messages can contain any human readable format.

The first digit specifies the category:

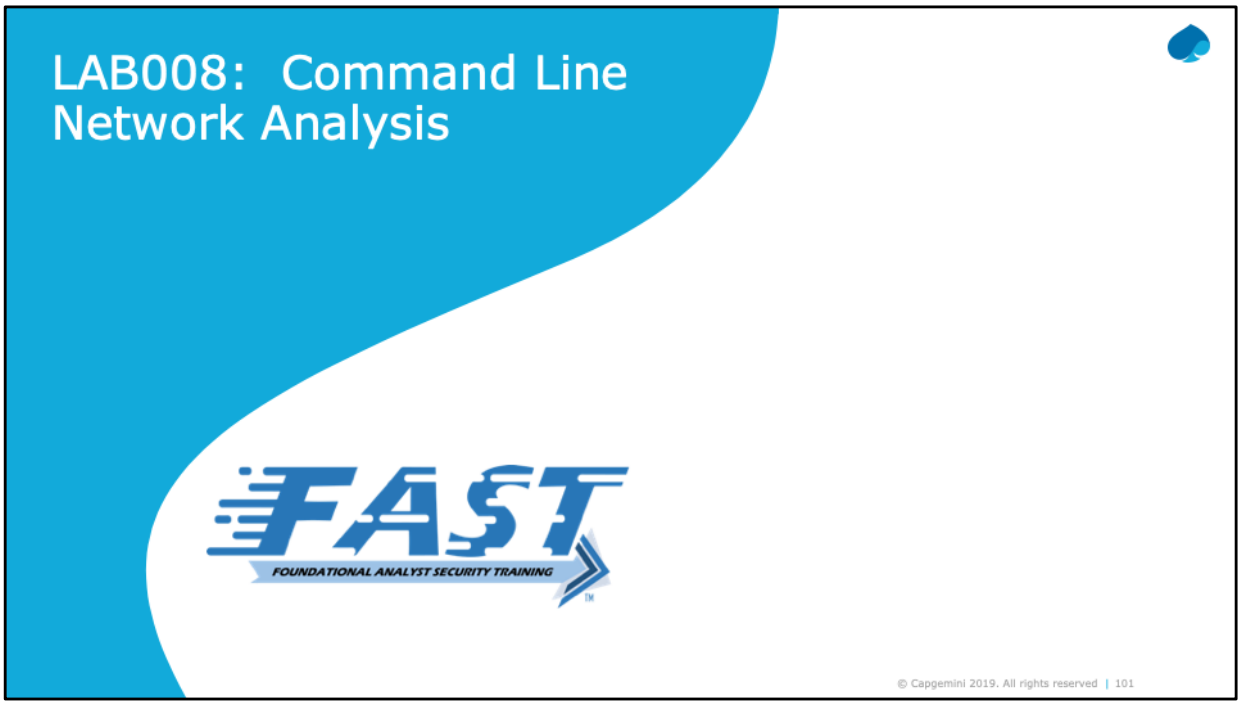
- 1XX Informational
- 2XX Success
- 3XX Redirection
- 4XX Client Error
- 5XX Server Error

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.

200 OK
Informational
The request has succeeded. The meaning of a success varies depending on the HTTP method:
201 Moved Permanently
Redirection
This response code means that the URI of the requested resource has been changed. Probably, 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.



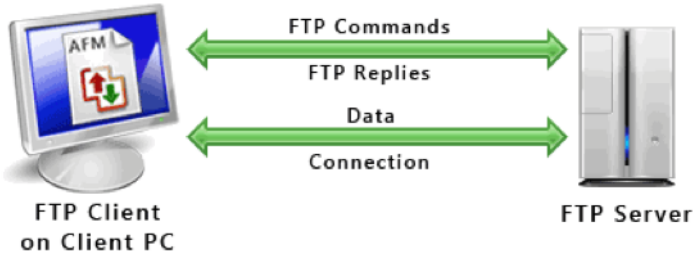
Content Source:

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



FAST Foundational Analyst Security Training

Why is FTP important to us?

- Delivery: The attacker must get the tools to the client. FTP is one good way
- Data exfiltration: the adversary is usually after data, and they have to get it out somehow

Port and Protocol are:

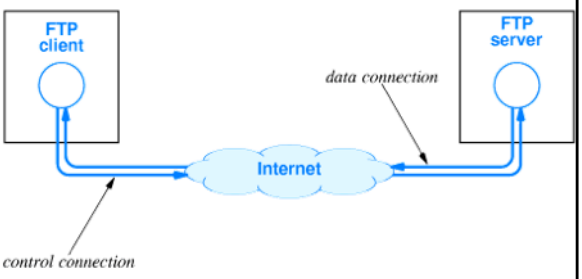
- Protocol: TCP
- Port: 20 (data), 21 (control)

Connection control is on Port 21; it remains open for the entire session. It is used to authenticate pass commands from client to server, and report status from server to client using codes, and it is clear text, 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



FAST Foundational Analyst Security Training

Connection Modes are:
- Active mode, which is rarely seen lately due to firewalls and NAT. The Client indicates desire to transfer file by sending IP address and port to server, and the client listens on specified port and waits for incoming server connection. The Server initiates TCP connection to client using specified address and port.
- Passive mode, where the client indicates desire to transfer by sending "PASV" command to server. The Server replies with IP address and port (usually port 20). The Client initiates TCP connection to server on specified IP address and port.

Transfer Modes are:
- ASCII mode, where data is transferred using ASCII formatted text streams in the format of the OS. This will not work for binary file types.
- Binary mode, which was formally called "Image" type. It 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

FAST Foundational Analyst Security Training

FTP still plays a large role in normal business processes.

FTP is a clear text protocol and is easily inspected. The adversary knows this, and instead uses high-levels of encryption to evade detection (e.g., compressing data in .RAR encrypted containers).

There are multiple FTP-like encrypted transfer mechanisms
- FTPS: FTP over SSL (aka FTP Secure)
- SFTP: 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)

FAST Foundational Analyst Security Training

Email is often an issue because Email is a common vectors for APT delivery and Users commonly use email.

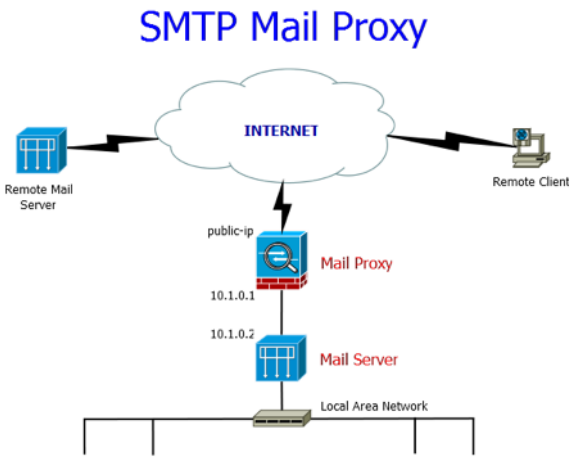
Common Types are:
- Internet Message Format (IMF) and Simple Mail Transfer Protocol (SMTP)
- WebMail
- MS Exchange and Messaging Application Programming Interface (MAPI)

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

Port and Protocol are:
- Protocol: TCP
- Port: 25 & 587

Message routing is depicted on the slide.

A key security consideration is that there are multiple security controls that help address email vulnerabilities; however, none are individually sufficient. The most robust defense is through layering and redundancy.

Simple Mail Transfer Protocol (SMTP) is based on end-to-end message delivery. An Simple Mail Transfer Protocol (SMTP) client contacts the destination host's Simple Mail Transfer Protocol (SMTP) server on well-known port 25, to deliver the mail. The client then waits for the server to send a 220 READY FOR MAIL message. Upon receipt of the 220 message, the client sends a HELO command. The server then responds with a "250 Requested mail action okay" message.

After this, the mail transaction will begin with a MAIL command that gives the sender identification as well as a FROM: field that contains the address to which errors should be reported.

After a successful MAIL command, the sender issues a series of RCPT commands that idenclfy recipients of the mail message. The receiver will the acknowledge each RCPT command by sending 250 OK or by sending the error message 550 No such user here. After all RCPT commands have been acknowledged, the sender issues a DATA command to inform the receiver that the sender is ready to transfer a complete mail message. The receiver responds with message 354 Start mail command with an ending sequence that the sender should use to terminate the message data. The termination sequence consists of 5 characters: carriage return, line feed, period, carriage return, and line feed (<CRLF>.<CRLF>).

The client now sends the data line by line, ending with the 5-character sequence <CRLF>.<CRLF> line, upon which the receiver will acknowledge with a 250 OK, or an appropriate error message if anything went wrong.
After the sending is completed, the client can follow any of these actions.

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

IMF is defined in RFC 5322. It is similar in some ways to HTTP.

Messages are streams of ASCII text separated by Carriage Returns. They contain a Header and Body, and they are heavily reliant on MIME types.

Headers are Pairs of field headings and values separated by ":". Records are separated by CRLFs, and Header blocks are terminated by two CRLFs.

The Body is lines of ASCII text. As with HTTP, advanced functionality (e.g., HTML, RichText, etc.) is provided by MIME specs.

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

Foundational Analyst Security Training

This slide depicts IMF commands and server responses.

Managed by IANA. For definitive list:

Permanent Headers: http://www.iana.org/assignments/message-headers/perm-headers.html

Provisional Headers: http://www.iana.org/assignments/message-headers/prov-headers.html

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

Trace Headers:

- Provide an invaluable trail of message handling.
- According to SMTP spec, EVERY server in the transport process MUST insert trace messages at the beginning of the message.
- Also, SMTP servers MUST NOT change or delete a received line.
- Return-Path: Added by the final system that delivers the message to the Recipient. Not to be confused with the reply-to field (which is added by the message originator).
- Received: Added by each system involved in the transport of the message. Actual data is optional. Might include info such as sending and receiving hosts, timestamps, VIA (may contain information about the transmission network), and WITH (protocol information).

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

FAST Foundational Analyst Security Training

MIME is an Internet standard initially designed to extend the format of email. However, it is also heavily used elsewhere.

It is the de facto format standard for email.

SMTP and IMF are ASCII protocols. MIME extends the capabilities to allow other types of information to be transmitted and processed.

Headers contain the Version, Content-ID which is usually used to identify multi-part messages; gives a globally unique identifier for each part, and the content-Type: Internet media type; managed by IANA; examples "application/pdf," "image/jpeg," "text/plain".

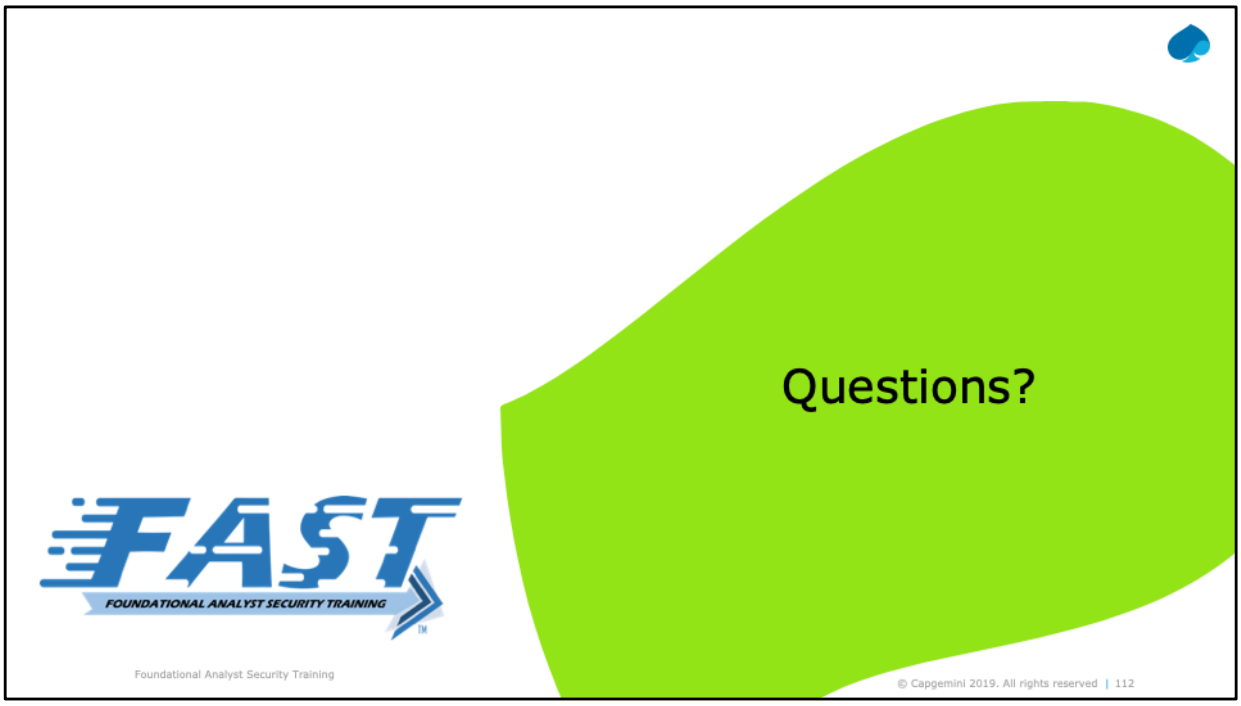
Content-Disposition, which defines the way MIME data is presented; two main types -"inline" and "attachment".

Content-Transfer-Encoding, which indicates that a binary-to-text encoding mechanism has been used (e.g. Base64).

Encoded-Word, which is used to identify non-ASCII header names and values.

Network Protection

Capgemini


Questions?
FAST
FOUNDATIONAL ANALYST SECURITY TRAINING
Foundational Analyst Security Training
© Capgemini 2019. All rights reserved | 112

Overview of Firewalls Types

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

FAST Foundational Analyst Security Training

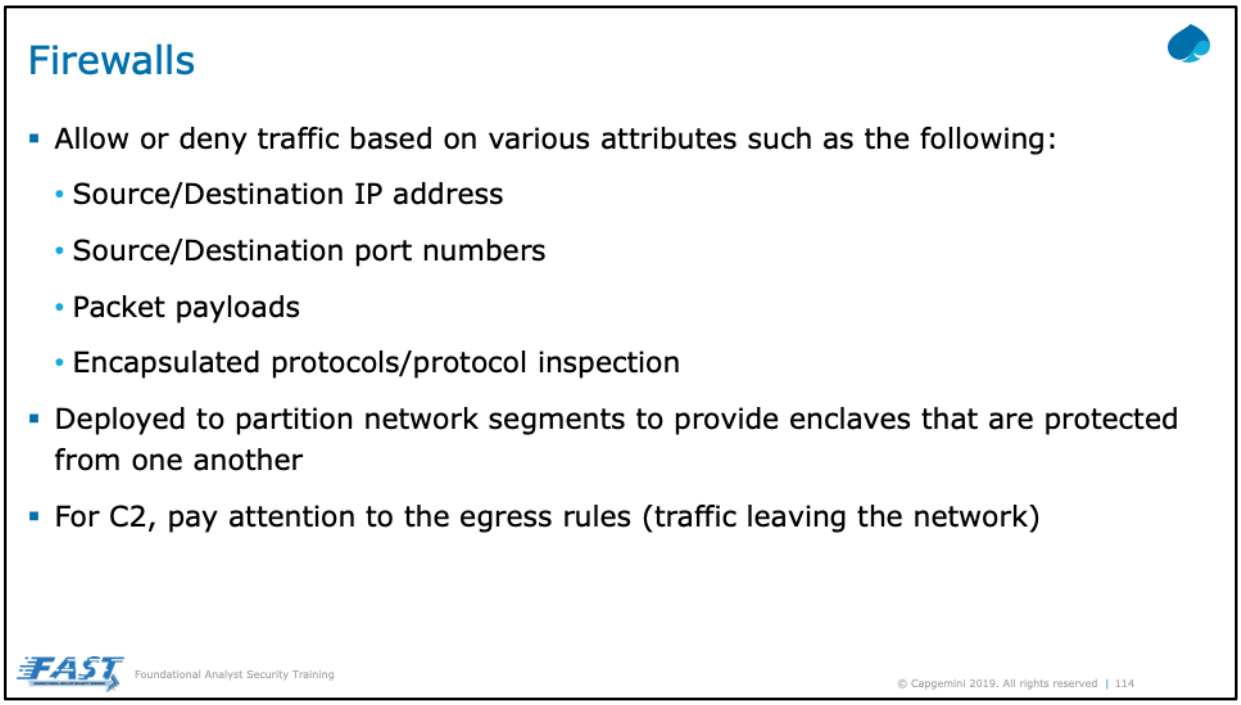
Packet Filtering, which will filter traffic based on information in the packets (routers and switches typically conduct packet filtering). Filters based on information, such as the allowed IP addresses, packet type, port number, etc.

Circuit-Level, which determines legitimate sessions b monitoring the TCP handshakes across the network.

Stateful Inspection, which are state-aware devices will not only examine each packet, but also keep track of the TCP session status and information (such as number of half-open sessions).

Application-Level, which filter packets for the service for which they are intended and other characteristics, such as the HTTP request string. These are often Proxies.

Next-Gen Firewalls , which typically combine packet inspection with stateful inspection, but also includes deep packet inspection.



Firewalls will allow or deny traffic based on various attributes, such as:

- Source/destination IP address
- Source/destination port numbers
- Packet payloads
- Encapsulated protocols/protocol inspection

They are commonly 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

Foundational Analyst Security Training

Proxy servers are traditional network perimeter devices (routers, firewalls) operated at layer 2 and 3. Since web protocols exist at layer 4, early network perimeter devices had no visibility into the content traversing the network boundaries.

Although modern perimeter devices are much more intelligent and can perform logging and inspection at the upper layers, this often increases load at critical network chokepoints. Analyzing this intricate traffic is outside the capabilities of most perimeter devices.

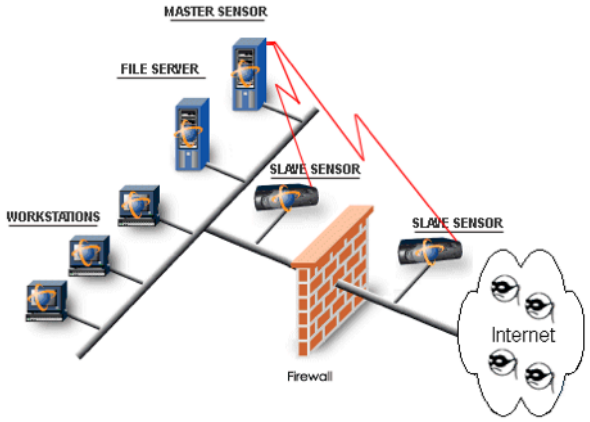
Proxy Servers provide multiple useful capabilities

- Filtering: Administrative control over what can or cannot traverse the perimeter. Accomplished via blacklists/whitelists, URL regex filtering, MIME-type filtering, or content association
- Caching: Stores a local copy of web content in order to reduce Internet bandwidth consumption and traffic congestion
- DNS Proxy: Consolidates and forwards internal DNS requests to external name servers
- Logging and Monitoring: Since the proxy server handles all outgoing web requests, it is an excellent place to monitor Internet usage for security and compliance

- Advanced functionality can include SSL Decryption, Real-time AV Scanning, Data Loss Prevention, and Reputation Analysis. Since most proxies have robust logging capabilities, Internet usage can be captured and attributed to specific systems and users

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



FAST Foundational Analyst Security Training

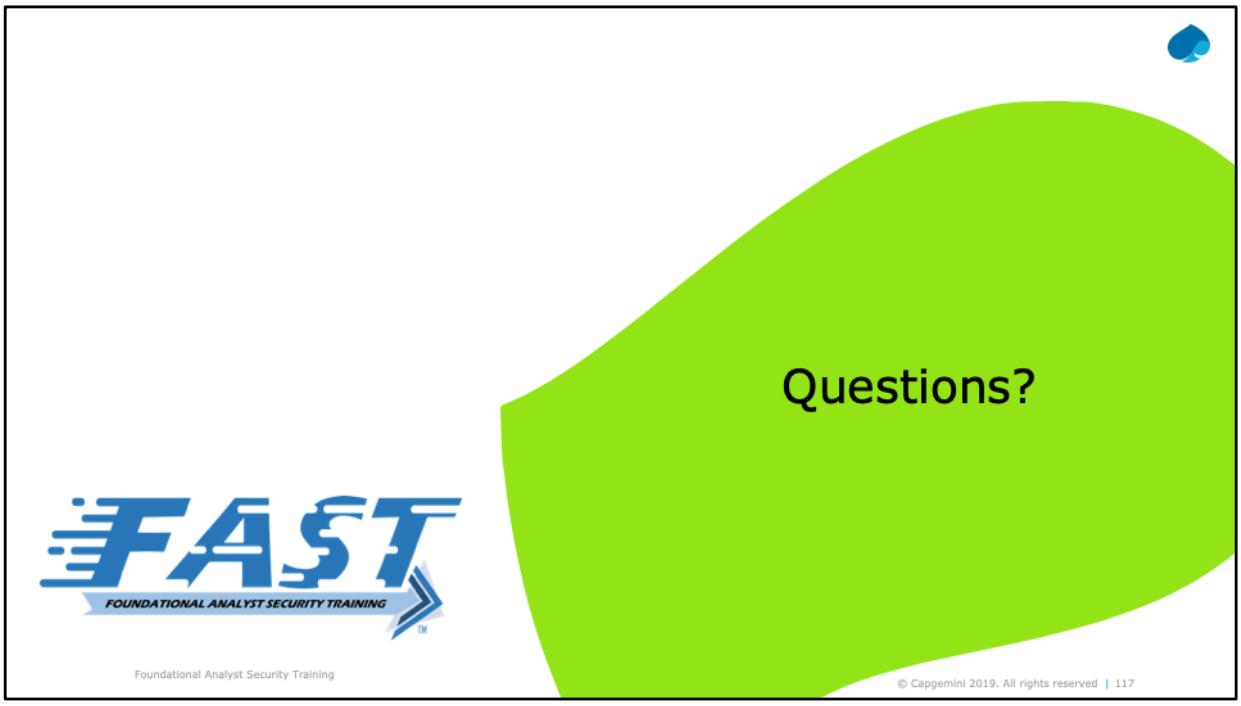
Intrusion detection systems (IDS) can be classified into different ways. The major classifications are Active and passive IDS, Network Intrusion detection systems (NIDS) and host Intrusion detection systems (HIDS).

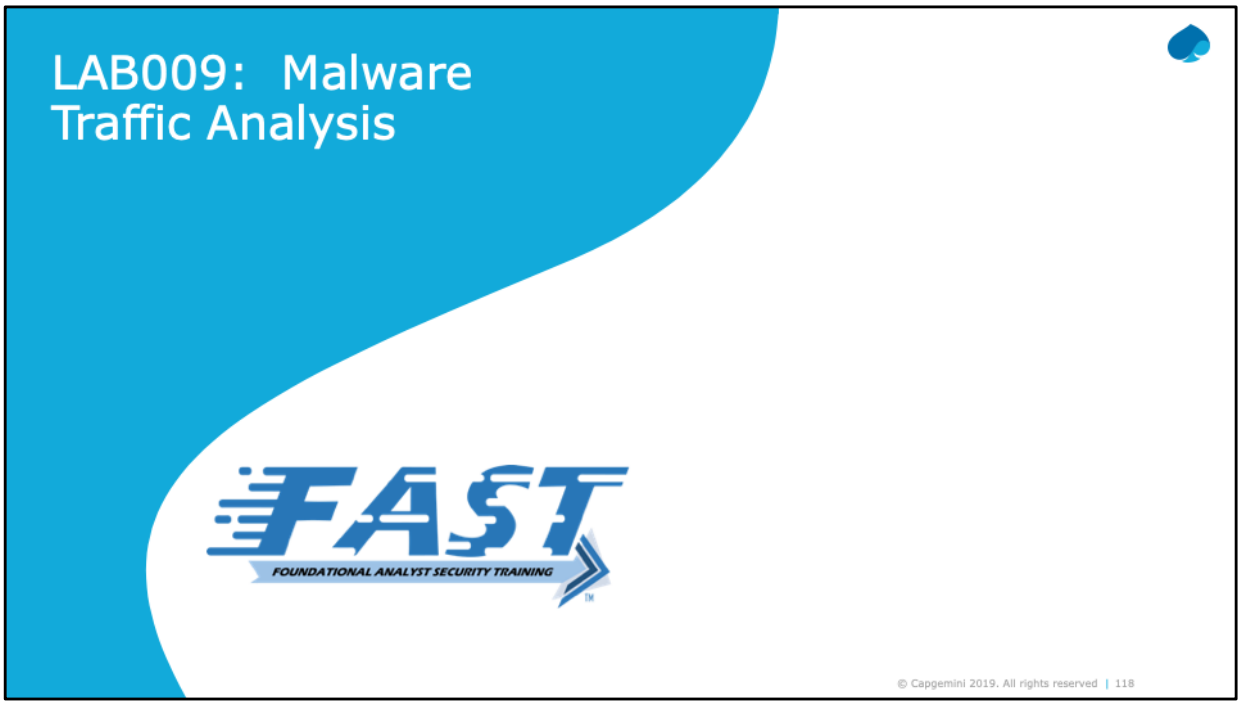
An active Intrusion Detection Systems (IDS) is also known as Intrusion Detection and Prevention System (IDPS). Intrusion Detection and Prevention System (IDPS) is configured to automatically block suspected attacks. A passive IDS monitor and alerts of potential vulnerabilities and attacks.

Network Intrusion Detection Systems (NIDS) usually consists of a network sensor (with a NIC) operating in promiscuous mode and a separate management interface. It is typically placed along a network boundary. A Host Intrusion Detection Systems (HIDS) includes software (agents) installed on monitored devices.

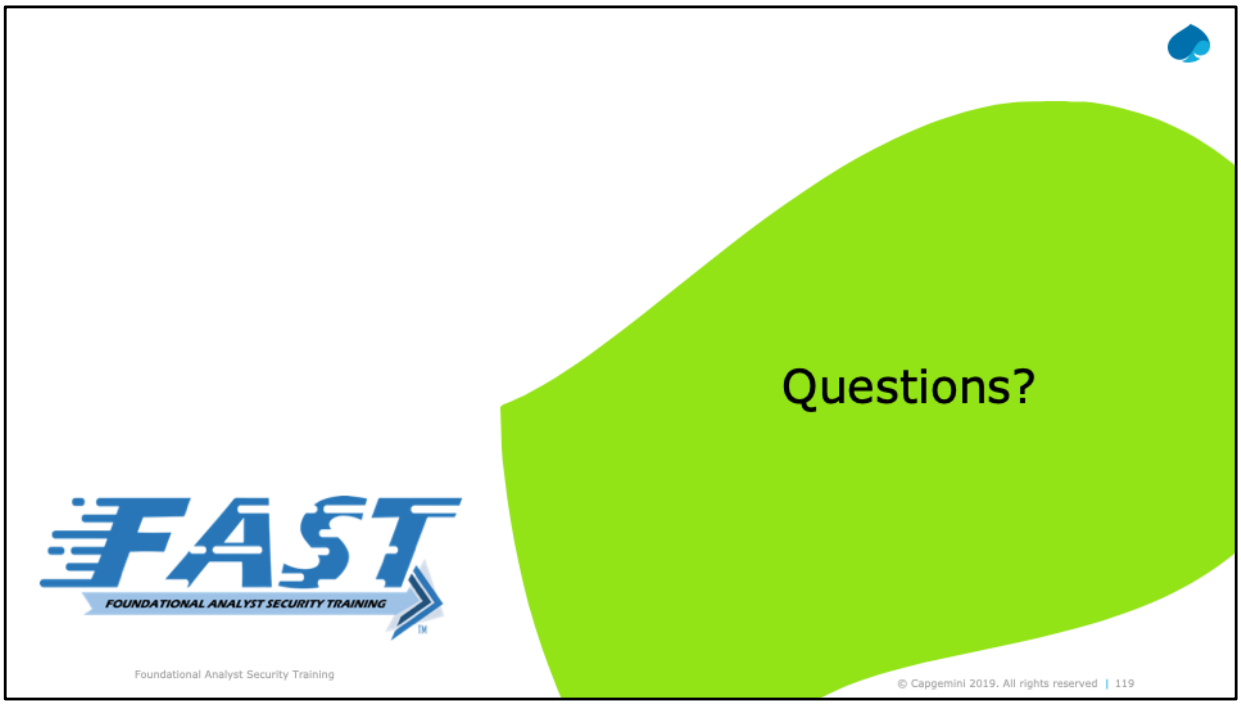
A knowledge-based (Signature-based) Intrusion Detection Systems (IDS) compares traffic with previous attack signatures and known system vulnerabilities. The primary disadvantages of Signature-based Intrusion Detection Systems (IDS) are signature database must be continually updated and maintained. A Behavior-based (Anomaly-based) Intrusion Detection Systems (IDS) compares traffic with a baseline of normal

system activity or patterns. These often generate more false alarms.


Questions?
FAST
FOUNDATIONAL ANALYST SECURITY TRAINING
Foundational Analyst Security Training
© Capgemini 2019. All rights reserved | 117



Content Source:


Questions?
FAST
FOUNDATIONAL ANALYST SECURITY TRAINING
Foundational Analyst Security Training
© Capgemini 2019. All rights reserved | 119


FAST
FOUNDATIONAL ANALYST SECURITY TRAINING
People matter, results count.
This presentation contains information that may be privileged or confidential and is the property of the CapGemini Group.
Copyright © 2019 CapGemini. All rights reserved.
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