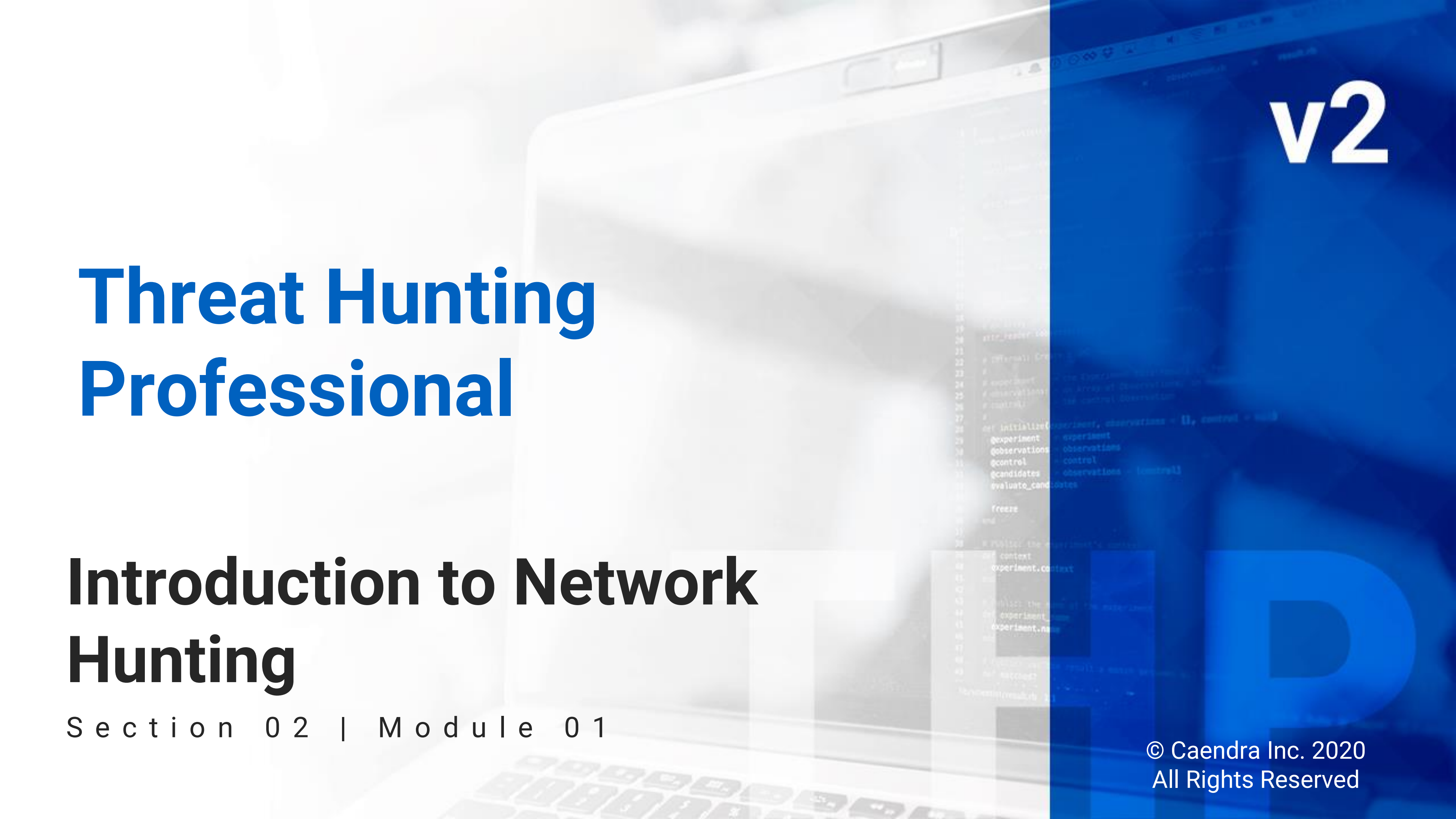


v2

Threat Hunting Professional

Introduction to Network Hunting

Section 02 | Module 01

Table of Contents

MODULE 01 | INTRODUCTION TO NETWORK HUNTING

1.1

Introduction

1.1 Introduction

In this section, we'll start looking through the eyes of a threat hunter who does not have threat intelligence (cyber threat intelligence or CTI) to aid him on what specifics to look for within the environment.

Note: Threat Intelligence will be referenced as CTI throughout the remainder of the course.

1.1 Introduction

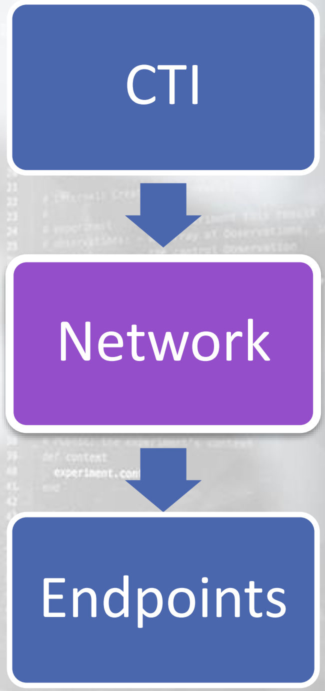
Within this course, we are helping you look at things from a top-down approach, meaning if you have CTI, then you'll be expected to use it to aid you in the hunt.

After using CTI to aid you in the hunt, where would you begin looking for threats? You'll look at the network through network analysis to find any signs of a threat.

1.1 Introduction

Threat Intelligence would be a great aid to us hunters, but we must not rely solely on CTI.

We will be looking at endpoints in the next section.



1.1 Introduction

Most small-medium sized businesses have an IT Security person or team and an Infrastructure/Network person or team as well. Unfortunately, in some cases, the IT Security “Team” is the Network “Team”.

In any case within this course, we’ll assume that the IT Security Team is not the same person as the Network Team.

1.1 Introduction

When it comes to Threat Hunters, they are typically part of a subset of the IT Security Team. In large organizations, a dedicated IR Team might be staffed. In smaller organizations, the hunter might be an on-call Incident Responder.

In most cases, the task of IR is committed to a 3rd party, and the hunter will typically be a Cybersecurity Analyst with a myriad of duties. Typically, these duties will not involve monitoring day to day network traffic or anything relating to the network.

1.1 Introduction

Again, the only time this is not the case is when the individual is a Jack, or Jane, of all trades, and he/she is responsible for multiple functions within the organization that would typically be broken up into different groups within the IT Team.

This would typically fall under smaller organizations.

1.1 Introduction

So let's build the scenario a threat hunter would encounter that would involve network hunting.

1. First, the Network Team would alert the hunter of unusual traffic on the network or in a specific subnet. Note: The organization doesn't have to go into full-blown IR mode at this point.

1.1 Introduction

2. Next, the Network Team or Threat Hunter would begin capturing packets.

3. And finally, the hunter would analyze the packet captures to confirm if there is an active threat in the network.

1.1 Introduction

How would the Network Team know if something suspicious is happening?

They will know through an alert from an appliance, such as an IDS/IPS. Other methods the Network Team can use to see that something odd is happening on the network is through statistical flow analysis (statistical modeling) and full packet capturing/analysis.

1.1 Introduction

Statistical Flow Analysis will provide visibility as to what is happening on the network. Graphs and advanced statistical analysis will aid the Network Team visually to see what is happening in the network historically (historical timeline) or in real-time.

So this, along with **network baselines** (if any), will allow the Network Team to see if anything is suspicious within the network, such as large unusual spikes, which can represent an exfiltration.

1.1 Introduction

We will not discuss in great detail the tools that will aid the Network Team with statistical flow analysis, network baselining or alerting.

Keep in mind that alerts can also arise within the SIEM and will also be visible to the IT Security Team. So, the notification can come from within the IT Security Team in addition to the Network Team.

1.1 Introduction

We will only be discussing tools that will aid the hunter with full packet logging in order to investigate further and to confirm the suspicion.

Before doing so, let’s briefly talk about TCP/IP, packets, network traffic, and network appliances.

1.2

TCP/IP & Networking Primer

1.2 TCP/IP & Networking Primer

Understanding how hosts communicate and more specifically, how they communicate within your environment is important because:

1. We need to know/understand the TCP/IP stack and normal network communication within the guides of the protocol. For example, with this knowledge, it will guide us to spot something out of the ordinary.

1.2 TCP/IP & Networking Primer

2. We also need to know/understand what a normal network communication is within the organization. For example, a forgotten workstation within the network sending data to an outside vendor via **FTP** every Saturday at midnight.

1.2 TCP/IP & Networking Primer

A Threat Hunter should be an experienced individual and should know/be familiar with the following:

- TCP/IP protocol stack and how it works
- Port numbers and typical applications that use those ports
- The normal behavior of typical applications as they transmit data through the corporate network

With that said, the following few slides will be a brief overview of TCP/IP & Networking that will serve as a refresher.

1.2 TCP/IP & Networking Primer

Most individuals in the IT field typically know and understand that information travels through the internal network and from network to network through ***packets***.

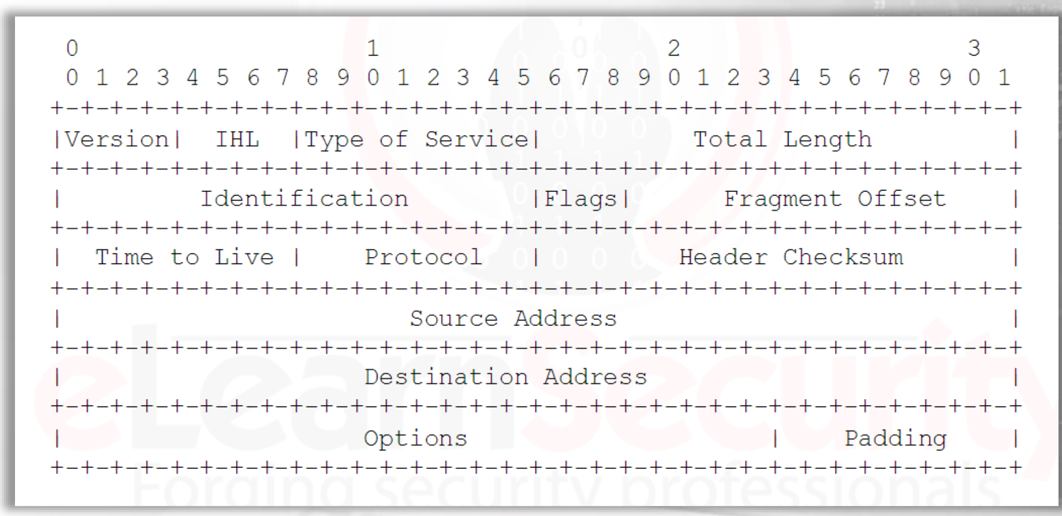
These packets are broken up at the source host and they are reconstructed at the receiving host. These packets contain a ***header*** which is followed by the ***payload***.

1.2 TCP/IP & Networking Primer

The header for each protocol has a specific structure; this will ensure that the receiving host can correctly interpret the payload and handle the overall communication.

1.2 TCP/IP & Networking Primer

For example, the IP protocol header is at least 160 bits (20 bytes) long, and it includes information to interpret the content of the IP packet.



```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version|  IHL  |Type of Service|          Total Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Identification        |Flags|      Fragment Offset    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Time to Live |    Protocol   |         Header Checksum       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Source Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination Address                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

1.2 TCP/IP & Networking Primer

The payload is the actual information to be sent to the destination host. This information can be part of an email or part of a file being downloaded/uploaded.

Understanding the structure of packets will aid the hunter when analyzing large packet captures, especially if it's a live capture. The hunter can search specific fields within the packet to narrow down the search or alert the hunter when the search criteria is met during a live capture.

1.2.1 OSI & TCP/IP Model

Different protocols operate at different layers of the OSI Model. The OSI Model consists of 7 layers and is used as a reference for the implementation of actual protocols.

You can find more information regarding the OSI model here.

https://support.microsoft.com/en-us/help/103884/the-osi-model-s-seven-layers-defined-and-functions-explained

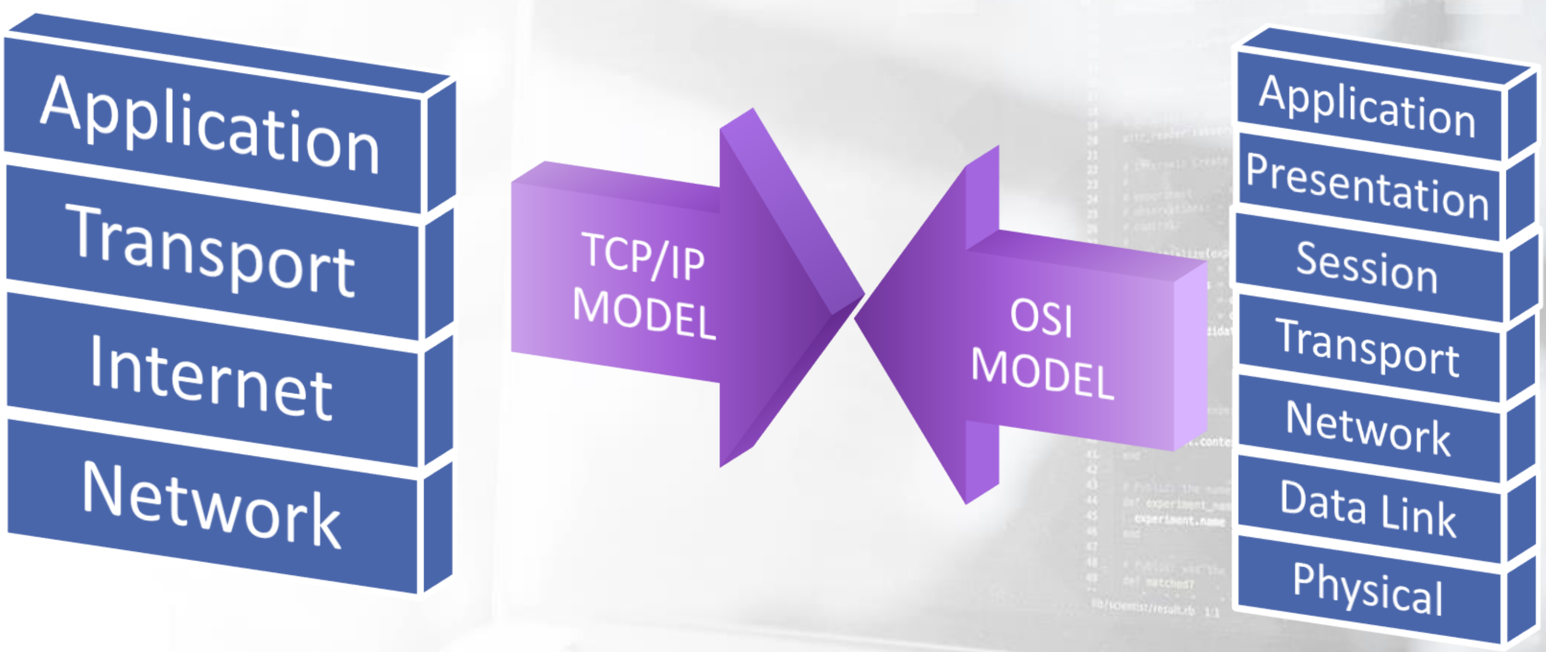
1.2.1 OSI & TCP/IP Model

The TCP/IP Model consists of only 4 layers instead of 7 layers like the OSI, but they mesh together even though it's fewer layers, and they are named differently, which you will see in the following slide.

You can read more about the TCP/IP and how it relates to the OSI model here.

https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-2000-server/cc958821(v=technet.10)

1.2.1 OSI & TCP/IP Model



1.2.1 OSI & TCP/IP Model

We need to remember that different protocols operate at different layers of the TCP/IP Model.

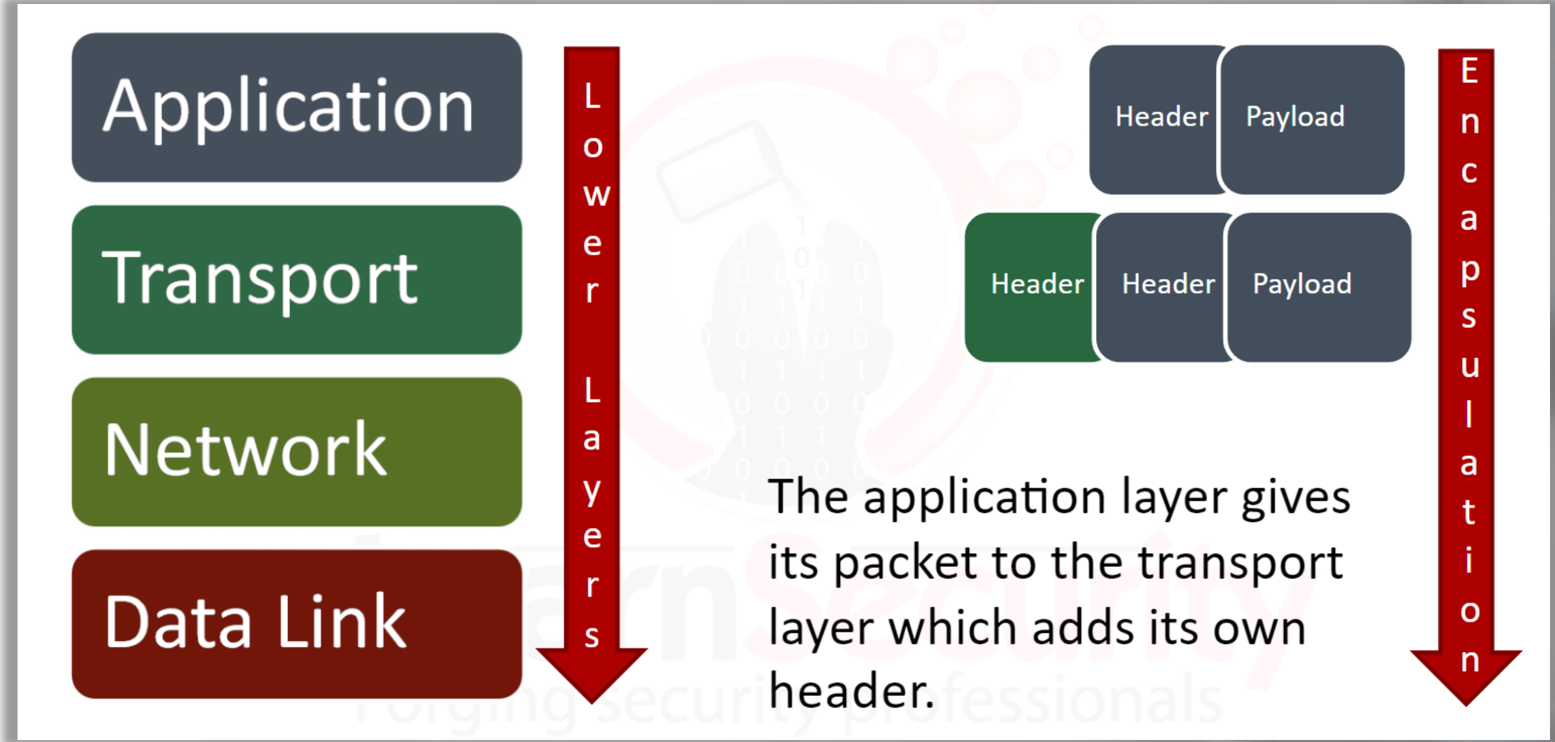
- Application Layer: FTP, SMTP, DNS, SNMP
- Transport Layer: TCP, UDP
- Internet Layer: IP, ARP, ICMP
- Network Layer: Ethernet, Token Ring, Frame Relay

As packets are broken up to be prepared to be sent to the destination host that each layer of the models will perform **encapsulation**.

1.2.2 TCP/IP Model

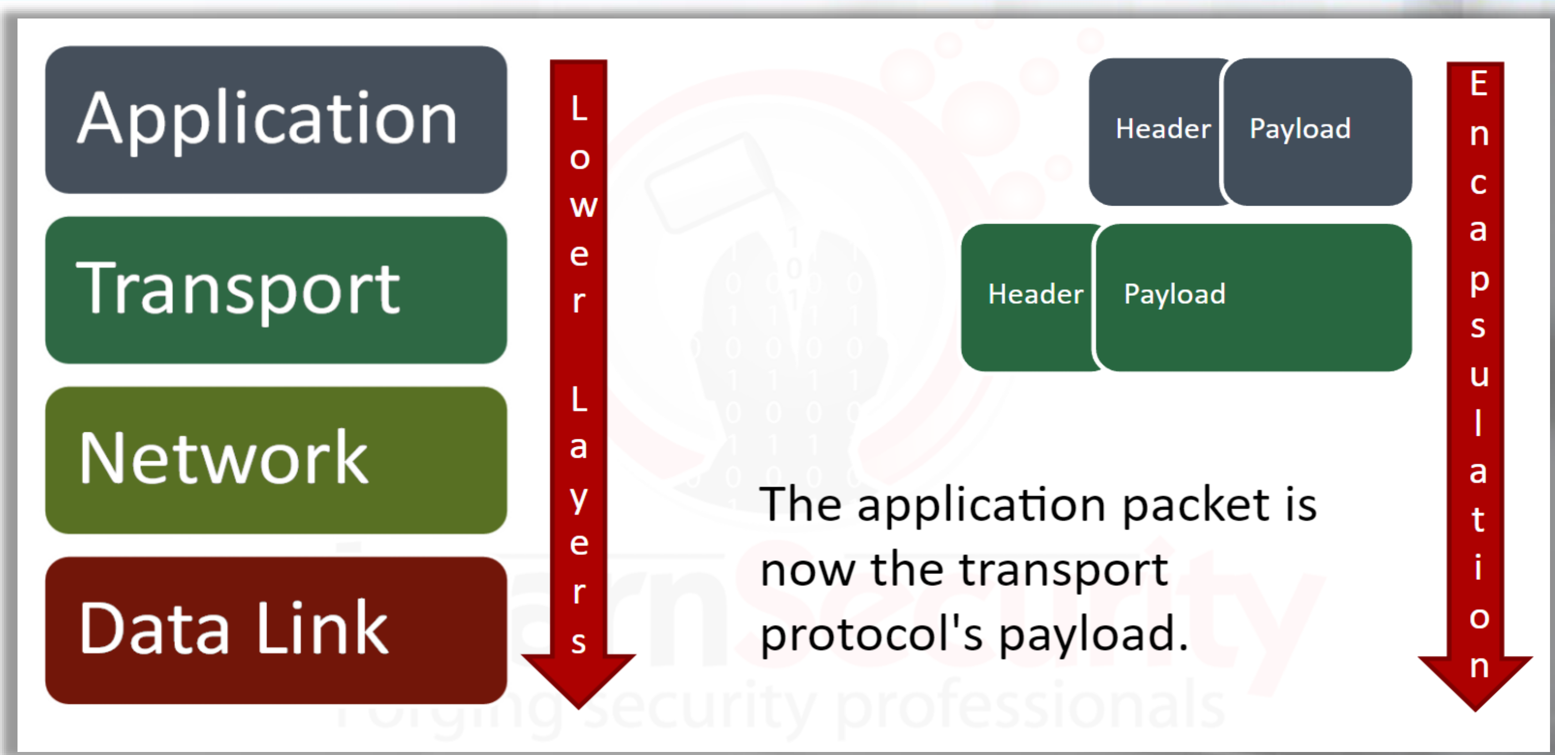
In the next few slides, we'll go through a visual reminder as to what encapsulation means and how it relates to the TCP/IP Model.

1.2.2 TCP/IP Model



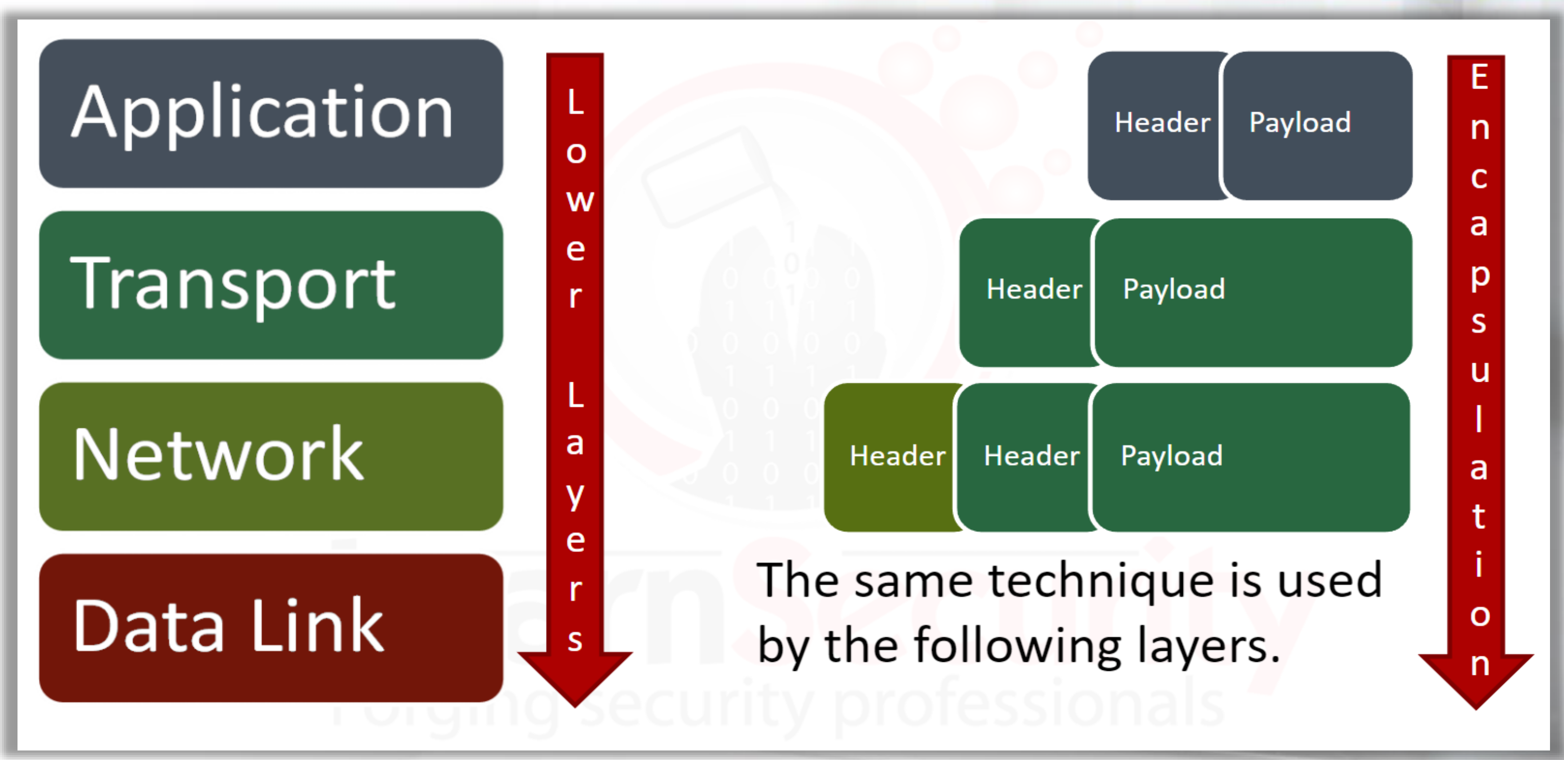
The application layer gives its packet to the transport layer which adds its own header.

1.2.2 TCP/IP Model



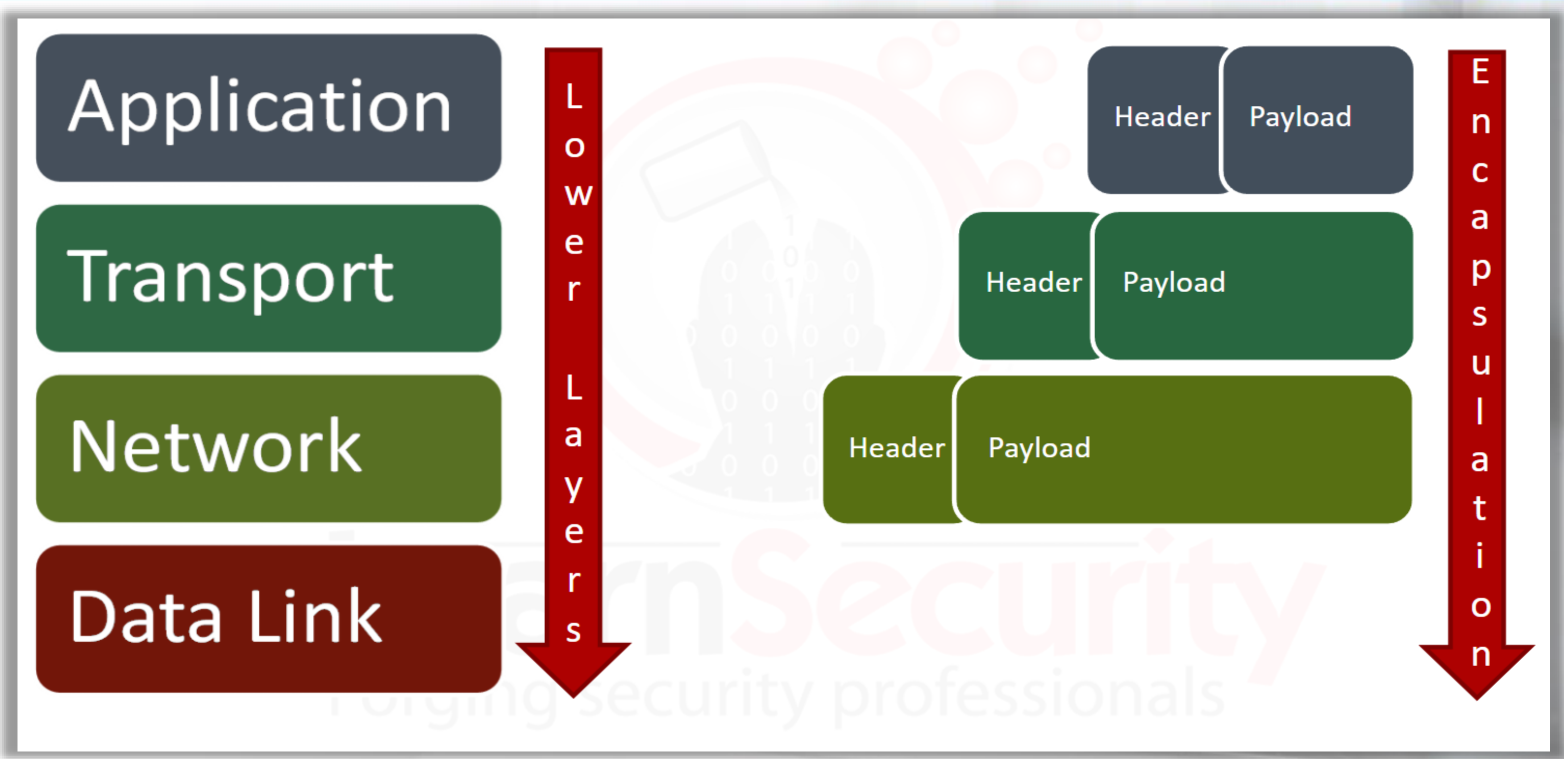
The application packet is now the transport protocol's payload.

1.2.2 TCP/IP Model

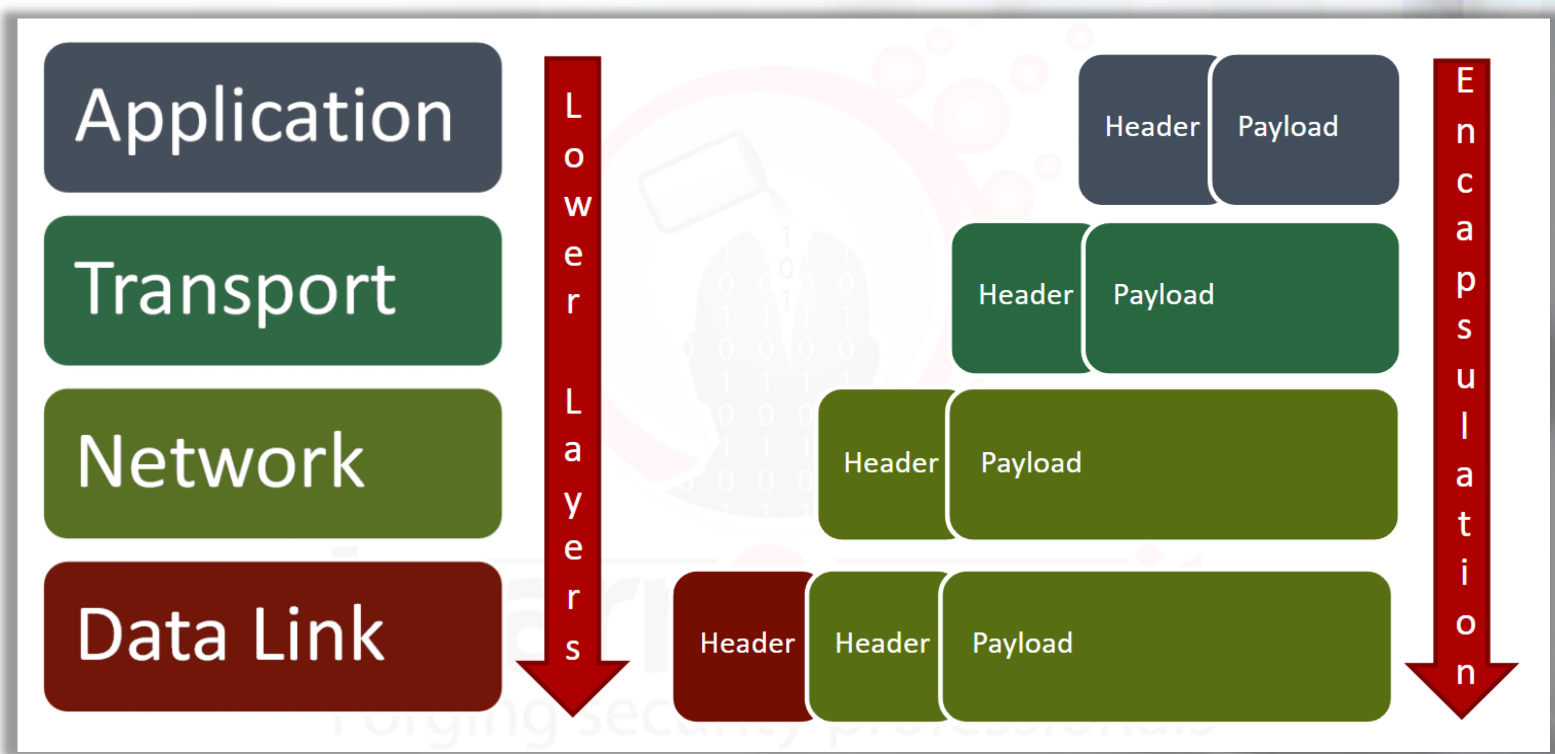


The same technique is used by the following layers.

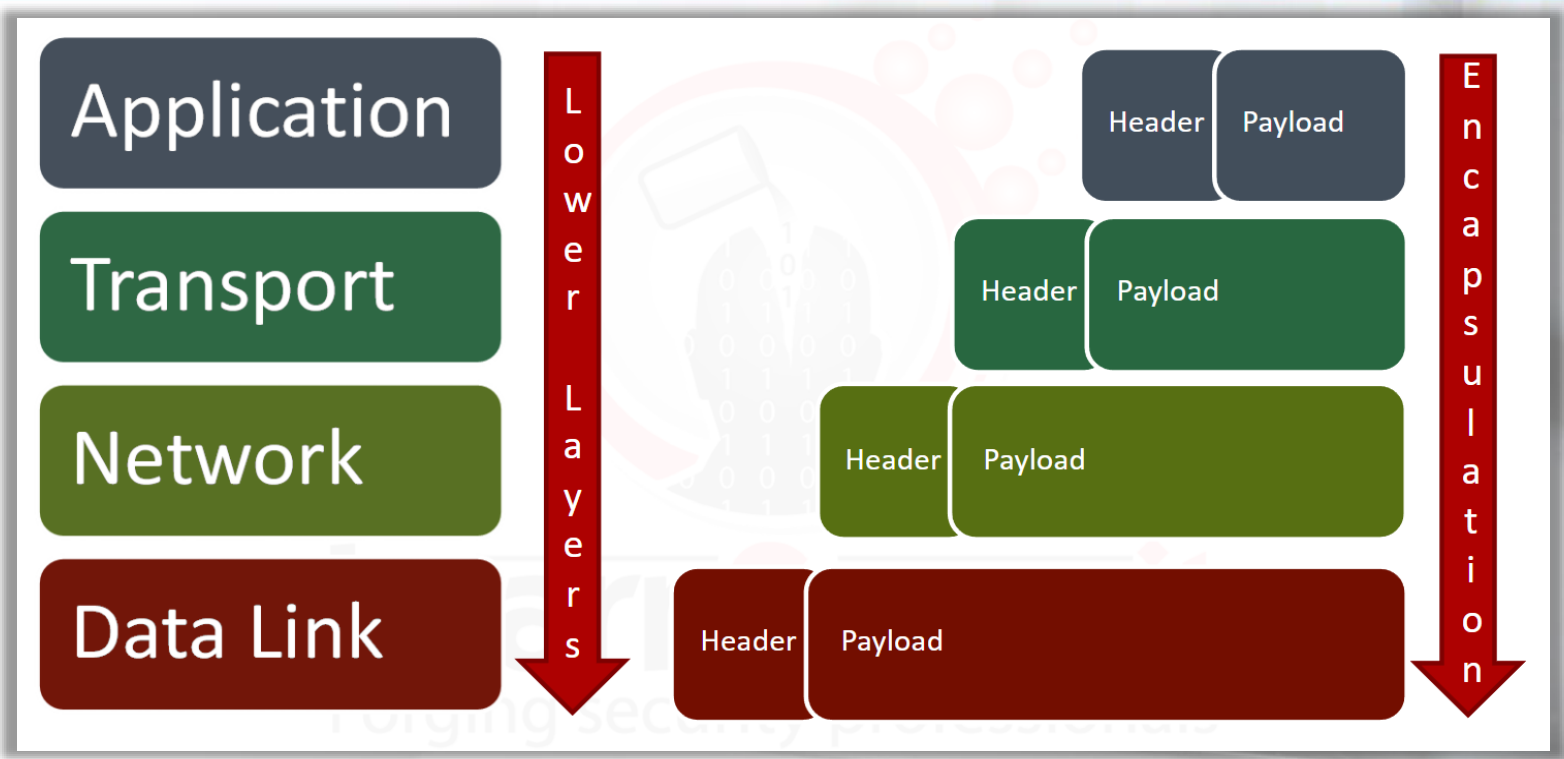
1.2.2 TCP/IP Model



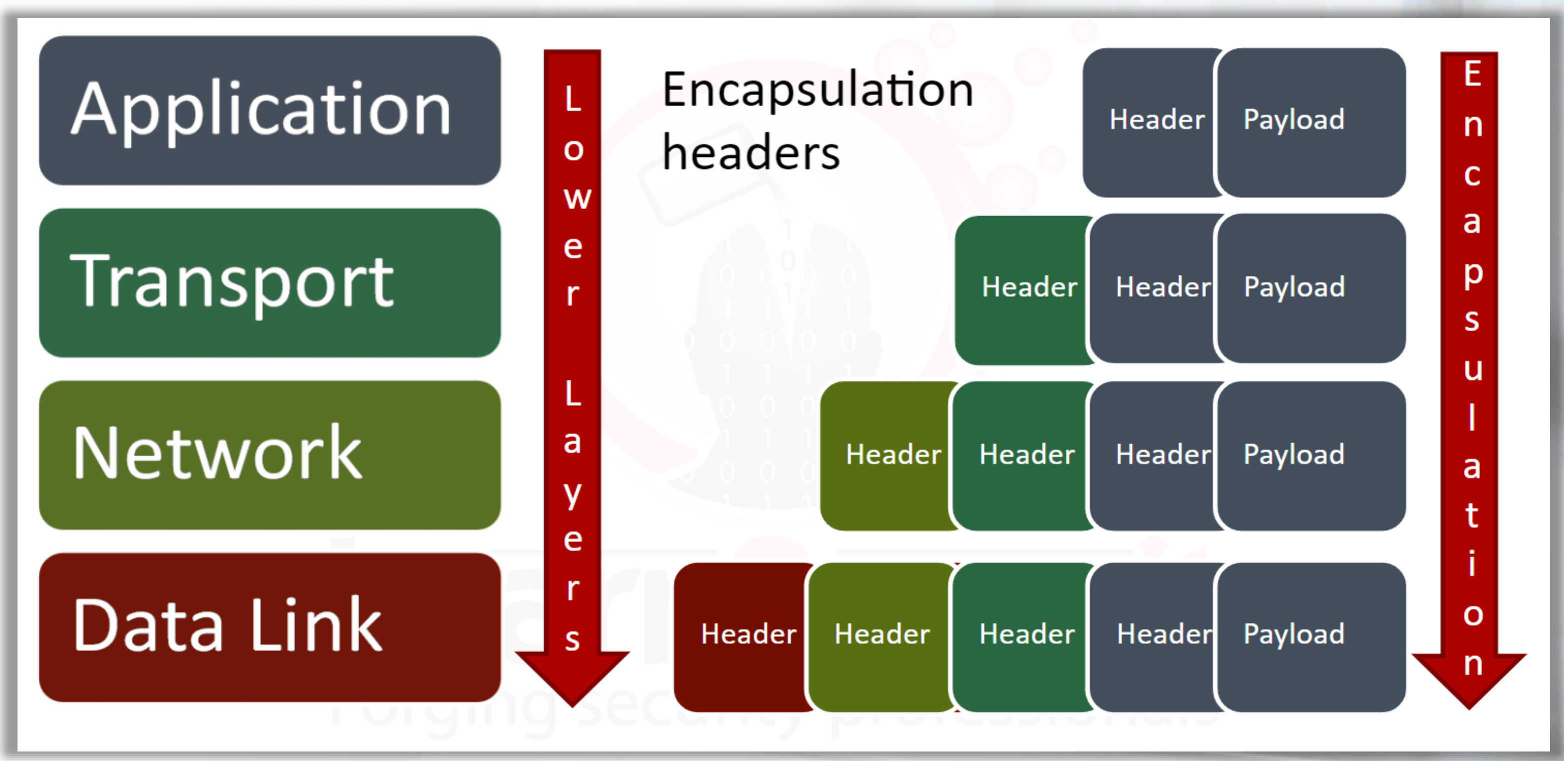
1.2.2 TCP/IP Model



1.2.2 TCP/IP Model



1.2.2 TCP/IP Model



1.2.2 TCP/IP Model

During encapsulation, every protocol adds a header to the packet, treating it as a payload; this happens to every packet.

At the destination host, which is receiving the packets, this process is done as well but in reverse order.

1.2.2 TCP/IP Model

The **Internet Protocol** (IP) is the protocol that runs on the **Internet** layer of the Internet Protocol suite, also known as TCP/IP.

IP is responsible for delivering the **datagrams** (IP packets are called datagrams) to the hosts involved in communication and uses IP addresses to identify a host.

1.2.3 Routers

Addressing devices is just half of the work needed to connect to a host. Your packets need to follow a valid **path** to reach it.

Routers are devices connected to different networks at the same time. They can forward IP datagrams from one network to another. The forwarding policy is based on **routing protocols**.

1.2.3 Routers

Routing protocols are used to determine the best path to reach a network. A router inspects the destination address of every incoming packet and then forwards it through one of its interfaces.

To choose the right forwarding interface, a router performs a lookup in the **routing table**. The routing table maps IP addresses to a specific interface. The table will also contain an entry with a **default address** (0.0.0.0); this is used when the router receives a packet whose destination is unknown.

1.2.3 Routers

As in the real world, there could be more than one way to reach a destination.

During path discovery, routing protocols also assign a **metric to each link**; this ensures that if two paths have the same number of hops, the fastest route is selected. The metric is chosen according to the channel’s estimated bandwidth and congestion.

1.2.4 Switches

In the same way that routers work with IP addresses, switches work with MAC addresses.

Switches have multiple interfaces, so they need to keep a forwarding table that binds one or more MAC addresses to an interface. The **forwarding table** is called the **Content Addressable Memory (CAM) table**.

1.2 TCP/IP & Networking Primer

In order to be comfortable analyzing network packets, you should be familiar with **ARP (Address Resolution Protocol)**, along with protocols such as **TCP (Transmission Control Protocol), UDP (User Datagram Protocol)**, and **DNS (Domain Name Service)** to name a few.

You should know how the protocol communicates as well as the similarities and differences between the protocols.

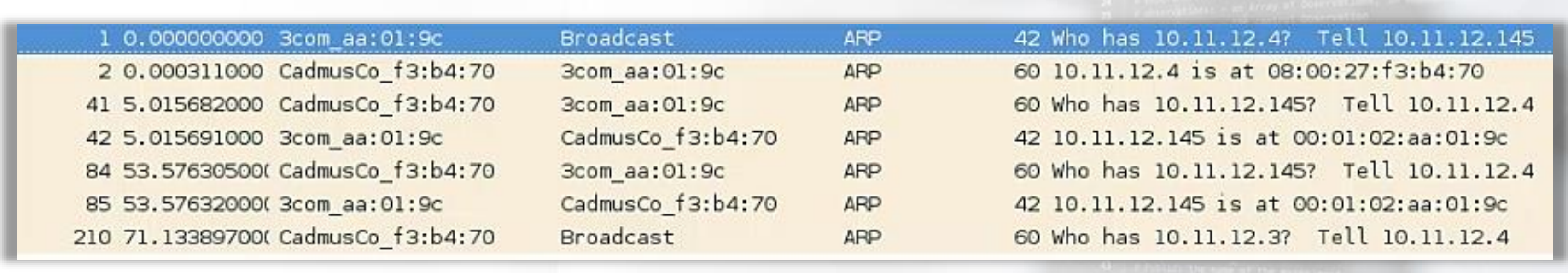
1.2.5 ARP Traffic

When a host *(A)* wants to send traffic to another *(B)*, and it only knows the IP address of *B*:

1. *A* builds an **ARP request** containing the IP address of *B* and FF:FF:FF:FF:FF:FF as the destination MAC address; this is fundamental because the switches will forward the packet to every host.
2. Every host on the network will receive the request.
3. *B* replies with an **ARP reply**, telling *A* its MAC address.

1.2.5 ARP Traffic

Here is an example of ARP request and ARP replies.



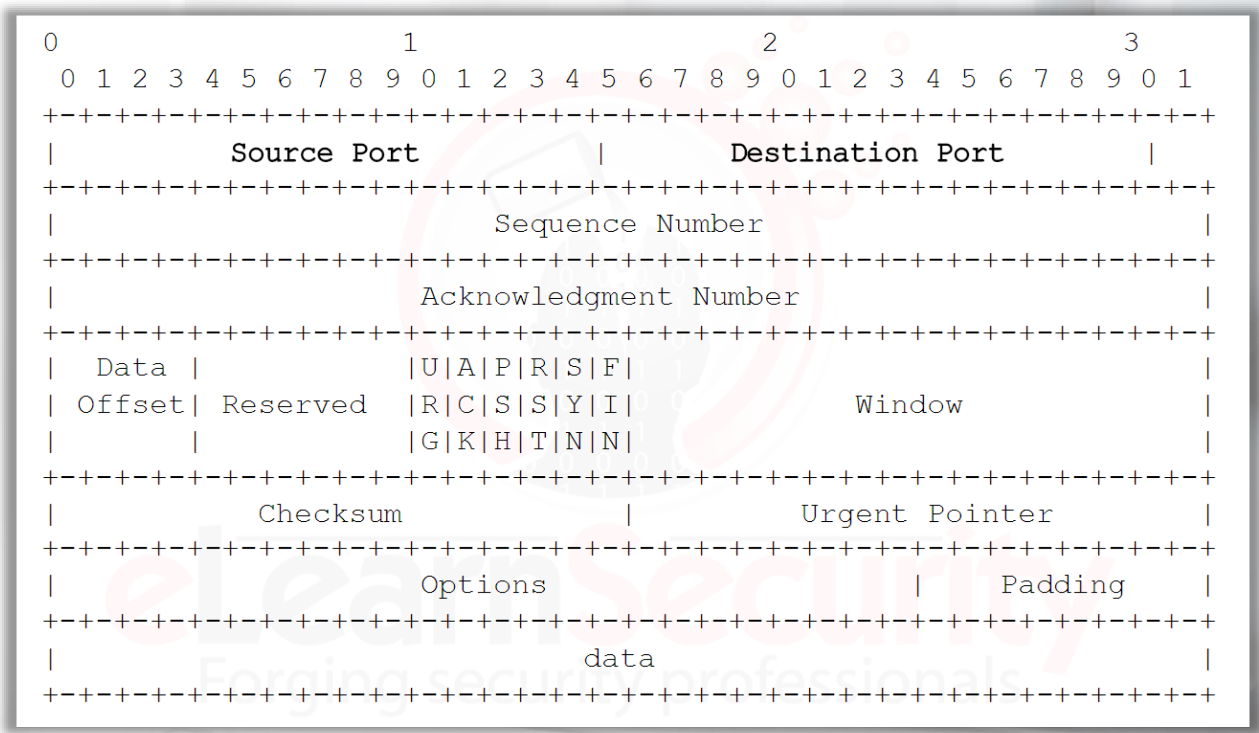
```
  1 0.000000000 3com_aa:01:9c       Broadcast           ARP       42 Who has 10.11.12.4?  Tell 10.11.12.145
  2 0.000311000 CadmusCo_f3:b4:70   3com_aa:01:9c       ARP       60 10.11.12.4 is at 08:00:27:f3:b4:70
 41 5.015682000 CadmusCo_f3:b4:70   3com_aa:01:9c       ARP       60 Who has 10.11.12.145?  Tell 10.11.12.4
 42 5.015691000 3com_aa:01:9c       CadmusCo_f3:b4:70   ARP       42 10.11.12.145 is at 00:01:02:aa:01:9c
 84 53.57630500( CadmusCo_f3:b4:70  3com_aa:01:9c       ARP       60 Who has 10.11.12.145?  Tell 10.11.12.4
 85 53.57632000( 3com_aa:01:9c      CadmusCo_f3:b4:70   ARP       42 10.11.12.145 is at 00:01:02:aa:01:9c
210 71.13389700( CadmusCo_f3:b4:70  Broadcast           ARP       60 Who has 10.11.12.3?  Tell 10.11.12.4
```

1.2.6 TCP Traffic

TCP uses a 3-way handshake to establish communication between two hosts because the protocol is connection-orientated.

1.2.6.1 TCP Header

Here is an example of a **TCP header**.

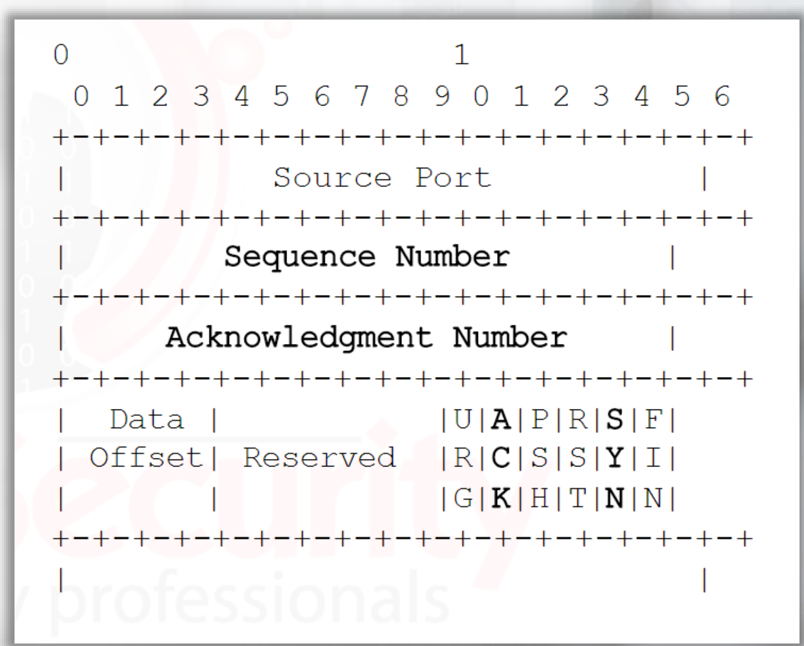


```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Acknowledgment Number                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Data |           |U|A|P|R|S|F|                               |
| Offset| Reserved  |R|C|S|S|Y|I|            Window             |
|       |           |G|K|H|T|N|N|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |         Urgent Pointer        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             data                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

1.2.6.1 TCP Header

The header fields involved in the handshake are:

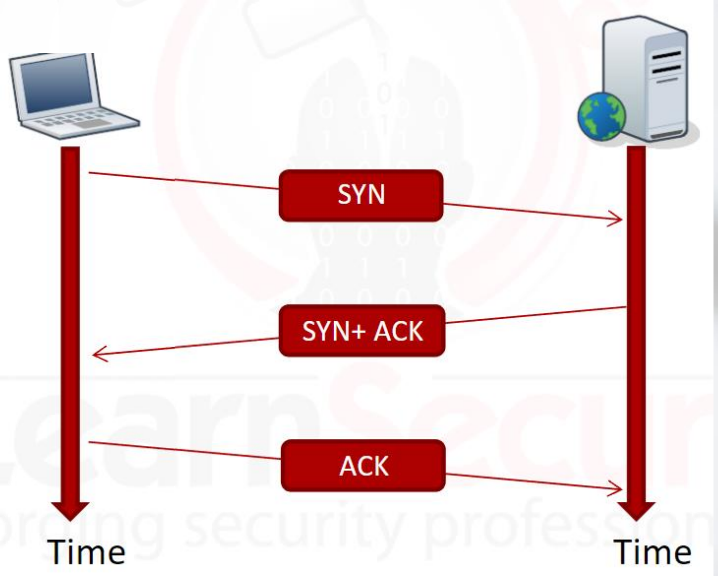
- Sequence number
- Acknowledgment number
- SYN and ACK flags



```
 0                   1
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |          Source Port              |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |       Sequence Number           |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |    Acknowledgment Number        |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |  Data |           |U|A|P|R|S|F|
 | Offset| Reserved  |R|C|S|S|Y|I|
 |       |           |G|K|H|T|N|N|
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                                 |
```

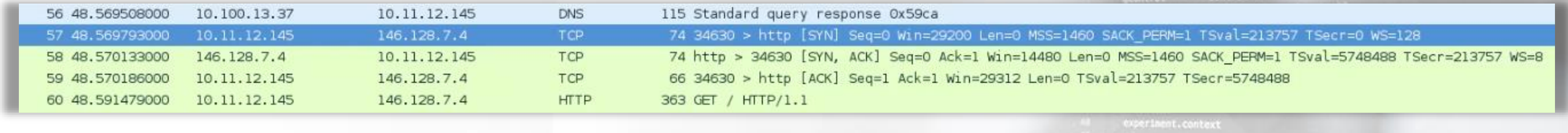
1.2.6 TCP Traffic

The steps in the handshake are used to synchronize the sequence and acknowledgment numbers between the server and the client.



1.2.6 TCP Traffic

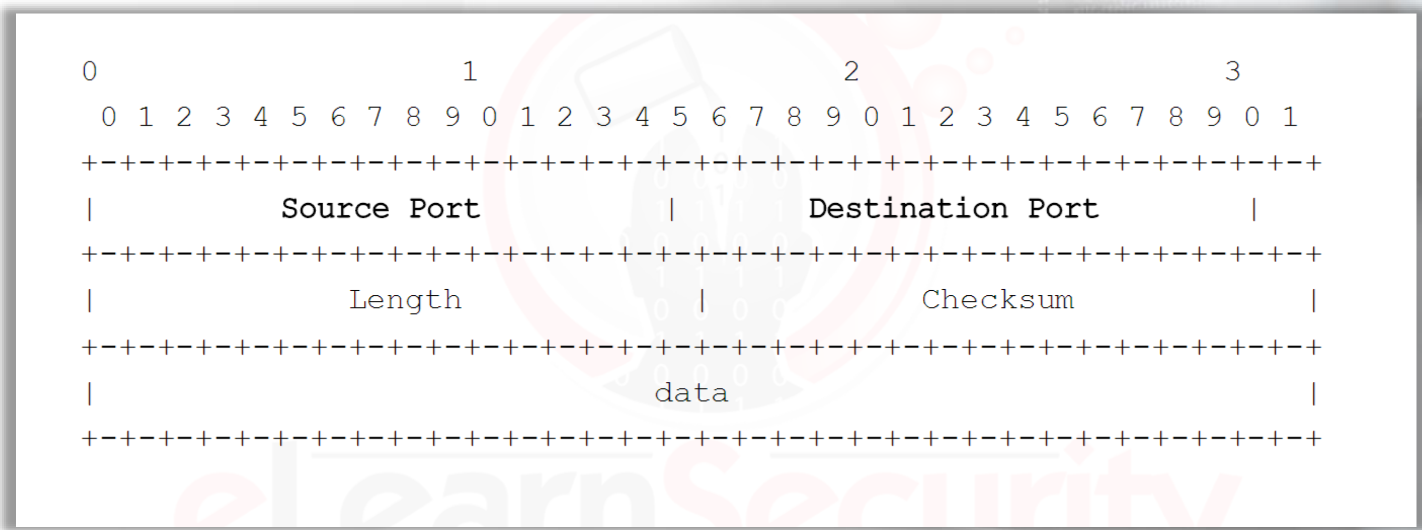
Below we can see the 3-way handshake through Wireshark.



```
56 48.569508000   10.100.13.37    10.11.12.145    DNS     115 Standard query response 0x59ca
57 48.569793000   10.11.12.145    146.128.7.4     TCP      74 34630 > http [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=213757 TSecr=0 WS=128
58 48.570133000   146.128.7.4     10.11.12.145    TCP      74 http > 34630 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460 SACK_PERM=1 TSval=5748488 TSecr=213757 WS=8
59 48.570186000   10.11.12.145    146.128.7.4     TCP      66 34630 > http [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=213757 TSecr=5748488
60 48.591479000   10.11.12.145    146.128.7.4     HTTP    363 GET / HTTP/1.1
```

1.2.7 UDP Header

Here is an example of a **UDP header**.



```
0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Source Port          |      Destination Port       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Length            |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             data                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

1.2.8 Common Ports

Lastly, each common protocol has a well-known port. Not only should you be familiar with common protocols, but you should also know which port they typically communicate on.

Ports are assigned by IANA and are referenced here.

http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml

1.2.8 Common Ports

Some common ports are:

- SMTP (25)
- SSH (22)
- POP3 (110)
- IMAP (143)
- HTTP (80)
- HTTPS (443)
- NETBIOS (137, 138, 139)
- SFTP (115)
- Telnet (23)
- FTP (21)
- RDP (3389)
- MySQL (3306)
- MS SQL Server (1433)

1.3

Packet Analysis & Tools

1.3 Packet Analysis & Tools

With full packet logging, we're capturing the raw packets as they traverse the network. It's the actual communication passed between nodes on the network.

As hunters, we can perform a live packet capture or obtain a PCAP (**packet capture**) file from the Network Team.

1.3 Packet Analysis & Tools

Packet captures are typically saved as a PCAP file, and nearly all packet capturing and analysis tools will export network traffic as a PCAP file and can also import PCAP files.

Other formats are available. Packet capturing and analysis tools can work with those formats, but PCAP is the standard format for packet captures.

1.3 Packet Analysis & Tools

Now, let's recall the typical scenario when a hunter can be called to implement network analysis.

- The Network Team is alerted or observes unusual traffic in the network or on a particular network segment.
- They notify the IT Security Team, and the hunter begins the hunt.
- The Network Team can provide a PCAP file to analyze, or the hunter can conduct a live packet capture.

1.3 Packet Analysis & Tools

Typically a hunter is not expected to scour through terabytes of network traffic to find something odd during scheduled hunts, and they are not expected to monitor traffic as it traverses the network.

Doing so is equivalent to standing on the corner of the busiest intersection during rush hour and trying to spot suspicious individuals.

1.3 Packet Analysis & Tools

That is the benefit of implementing defense-in-depth and creating a security monitoring program. Appliances should already be in place to catch or at least alert when suspicious activity occurs as long as the appliance and the ruleset are appropriately configured.

CTI can assist greatly with adding emerging threats to the ruleset.

1.3 Packet Analysis & Tools

One situation where the hunter might need to inspect traffic is when something occurs. For example, if the IDS/IPS appliance goes down for a certain amount of time, an analysis of the packets that traversed the network during that time would be inspected to see if any malicious traffic can be seen while the security appliances were/are down.

Again, this is not an everyday occurrence and might be a task that a hunter might have to undertake.

1.3 Packet Analysis & Tools

As a member of the IT Security Team, you are a native of the land (so to speak). That means you should be very familiar with the network infrastructure, different network segments, IP addressing scheme, any specific internal network rules, egress points, etc.

The same can't be said regarding an outside security consultant who comes in and knows nothing about the network.

1.3 Packet Analysis & Tools

You should know where to strategically tap into the network segment to begin a live network traffic capture.

If not, then that information and/or assistance needs to be obtained from the Network Team.

1.3 Packet Analysis & Tools

Typically in enterprise networks, the IT Security Staff use Windows machines. Red Teamers normally use Linux as their platform, whereas Blue Teamers normally use Windows as their primary platform.

As a Threat Hunter, you’re more of a Purple Teamer, so you must be familiar with both.

1.3 Packet Analysis & Tools

In this module, we'll briefly introduce you to some well-known tools used for capturing, filtering, and analyzing network traffic on Windows and Linux platforms.

In the next module, we'll use these tools and analyze different variations of network traffic.

1.3 Packet Analysis & Tools

You might be familiar with these tools, but we can’t assume that you are. Feel free to skip ahead if you are.

A good portion of IT Security Professionals are aware of certain tools and have had the occasion to use these tools, but that doesn’t necessarily mean they know how to use the tool in a real-life situation.

1.3 Packet Analysis & Tools

Before we begin looking at the various tools to capture network traffic, it is important to note a few things to consider during a **LIVE** network capture.

1. Capture and test to confirm that you're indeed capturing the traffic you intend to capture. This can help to prevent a situation where you are not capturing the packets you need.

1.3.1 Live Network Captures

2. Confirm that you have enough computing power to handle all the packets you’ll be capturing, especially in a heavy traffic network segment. You might think you’re capturing all packets, but you will only be capturing what your device can handle.

3. Make sure you have enough disk space on your device. It would be bad if you’re running a live network capture, and your device or VM only has a few GBs left of disk space. It will only get that large if you’re capturing data for an **extended** period of time.

1.3.1 Live Network Captures

4. Lastly, we don’t want to forget the fundamentals of network hardware, such as **switches**. Remember with switches that traffic destined for a particular host is point-to-point, even while technically on the same segment. To capture data, you will need to be connected to a **mirrored port**.

1.3.1.1 Port Mirroring

Port Mirroring is the process of replicating traffic destined for one or more ports to this specific port, **mirrored port**, which will be used for network analysis or, more specifically, packet analysis.

In the Cisco World, this port is known as the **Cisco Switched Port Analyzer** or a **SPAN** port.

1.3.1.1 Port Mirroring

The Network Team or Network Administrator would be responsible for providing this access to the Threat Hunter unless the Network Admin IS the Threat Hunter, then he/she would configure the switch accordingly based on the capabilities of the switch.

If the port mirroring option is not available, there are other options to sniff network traffic on the switch:

- Tapping the network cable
- MAC flooding (red team tactic)
- ARP spoofing (red team tactic)

1.3.1.2 Network Tap

Network Taps entail physically tapping the wire or network cable, whether it’s a copper cable or fiber.

Hardware is available that will allow us to tap the wire and intercept the traffic as it traverses the cable. An example of this is a **Vampire Tap**.

1.3.1.2 Network Tap

You can also use an **Inline Network Tap**. This type of tap is inserted ‘inline’ between two physical network devices, such as a firewall and a switch.

The Inline Network Tap would replicate copies of packets on a separate port, or ports, as it passes along the packets to its destination.

1.3.1.3 MAC Floods

MAC flooding is considered **active** sniffing and a tactic used by Red Teams; this should only be used when extremely necessary as you don’t want to cause unnecessary stress to the network equipment when you don’t need to. Authorization would be required from management to use this method.

MAC Flooding is meant to stress the switch and fill its CAM table. When the space in the CAM table is filled with fake MAC addresses, the switch can no longer learn new MAC addresses. The only way to keep the network alive is to forward the frames meant to be delivered to the unknown MAC address on all ports of the switch, thus making it fail open and act like a hub.

1.3.1.4 ARP Poisoning

ARP poisoning or ARP spoofing is another **active** technique but can be considered **stealthy**. It doesn't bring down the functionalities of the switch, as MAC Flooding does, but instead, it exploits the concept of traffic redirection. It is also considered a Red Team technique.

By exploiting the network via ARP Poisoning, we can redirect the traffic of the host(s) we want to monitor to our machine; this will allow us to monitor the traffic intended for the host(s) of interest. It is important to note that this technique is used to perform **Man in the Middle** attacks.

1.3 Packet Analysis & Tools

Remember that these options are only listed as alternatives if, for some reason, you can’t get hooked to a SPAN port or obtain network taps.

Again, you will need authorization from management to perform these tasks on the corporate network.

1.3.2 libpcap

Libpcap is a Unix C library that provides the capabilities for packet sniffing and analysis tools to capture and filter packets. **Wireshark** and **tcpdump** are libpcap-based tools.

Many other packet sniffing and analysis tools that are based on libpcap were created for special functions, such as **tcpflow** and **ngrep**. There are many libpcap-based tools, but we are only focusing on the most popular within this course.

1.3.2 libpcap

For Windows systems, **WinPcap** was created.

WinPcap is a libpcap library that was designed for Windows.

1.3.3 Wireshark

The first tool we'll look at is Wireshark. Wireshark is a network sniffer and a protocol analyzer; this means that you can use it to analyze every packet, traffic stream, or connection that hits your computer's network interface(s).

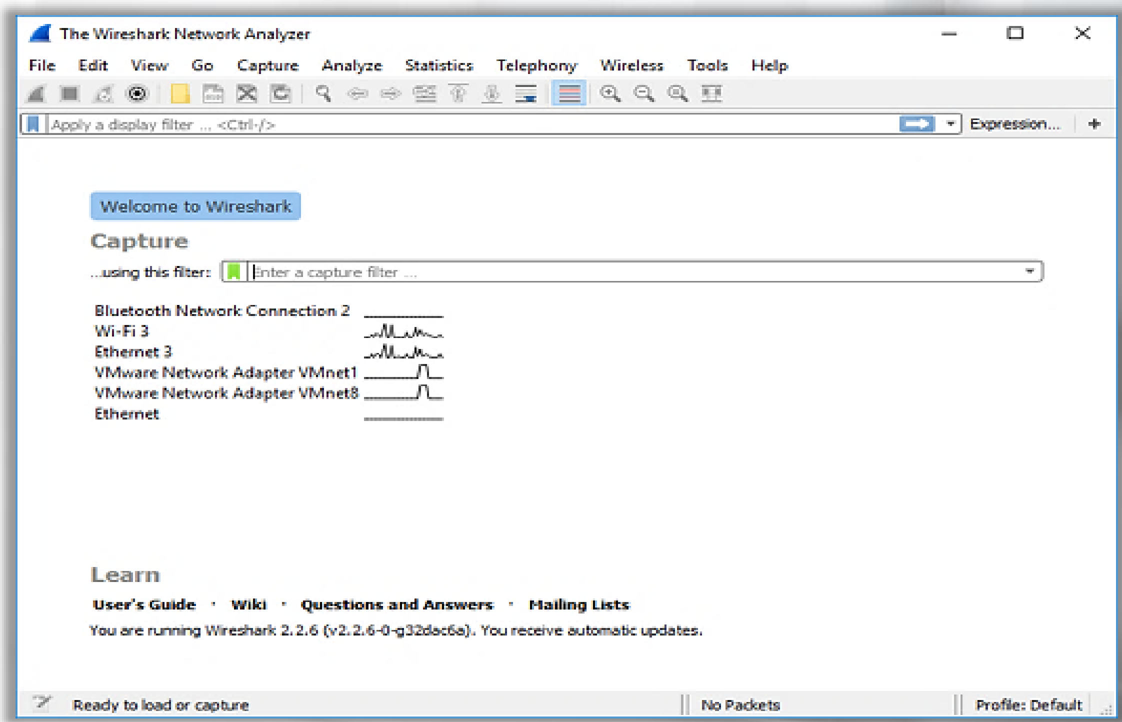
Wireshark is free software and can be installed on most modern operating systems. You can download Wireshark from here. In the following slides, we will briefly go over how to configure it so we can use its main features.

https://www.wireshark.org/download.html

1.3.3 Wireshark

Here we see the main window of Wireshark, where we can select the interface we'd like to use to capture network traffic.

We can also enter any filters to narrow down the specific traffic we would like capture.


The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
Welcome to Wireshark
Capture
...using this filter:
Enter a capture filter ...
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists
Ready to load or capture
No Packets
Profile: Default

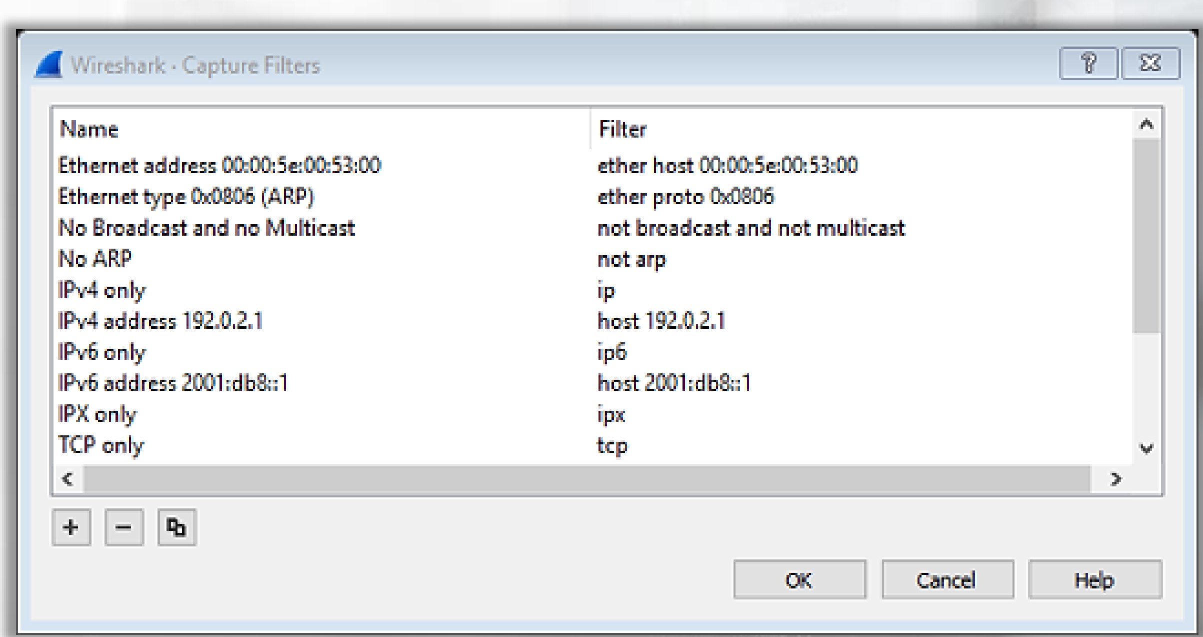
Bluetooth Network Connection 2
Wi-Fi 3
Ethernet 3
VMware Network Adapter VMnet1
VMware Network Adapter VMnet8
Ethernet

You are running Wireshark 2.2.6 (v2.2.6-0-g32dac6a). You receive automatic updates.

1.3.3 Wireshark

By selecting the green ribbon and Manage Capture Filters, we are taken to screen we see here.

On this screen, we can select a Capture Filter or create an entirely new one for later use.



Name	Filter
Ethernet address 00:00:5e:00:53:00	ether host 00:00:5e:00:53:00
Ethernet type 0x0806 (ARP)	ether proto 0x0806
No Broadcast and no Multicast	not broadcast and not multicast
No ARP	not arp
IPv4 only	ip
IPv4 address 192.0.2.1	host 192.0.2.1
IPv6 only	ip6
IPv6 address 2001:db8::1	host 2001:db8::1
IPX only	ipx
TCP only	tcp

1.3.3 Wireshark

You can also go to CAPTURE > OPTIONS on the menu bar to open a new window, Wireshark – Capture Interfaces.

In this window, we'll be able to configure more options such as Output and other Options.

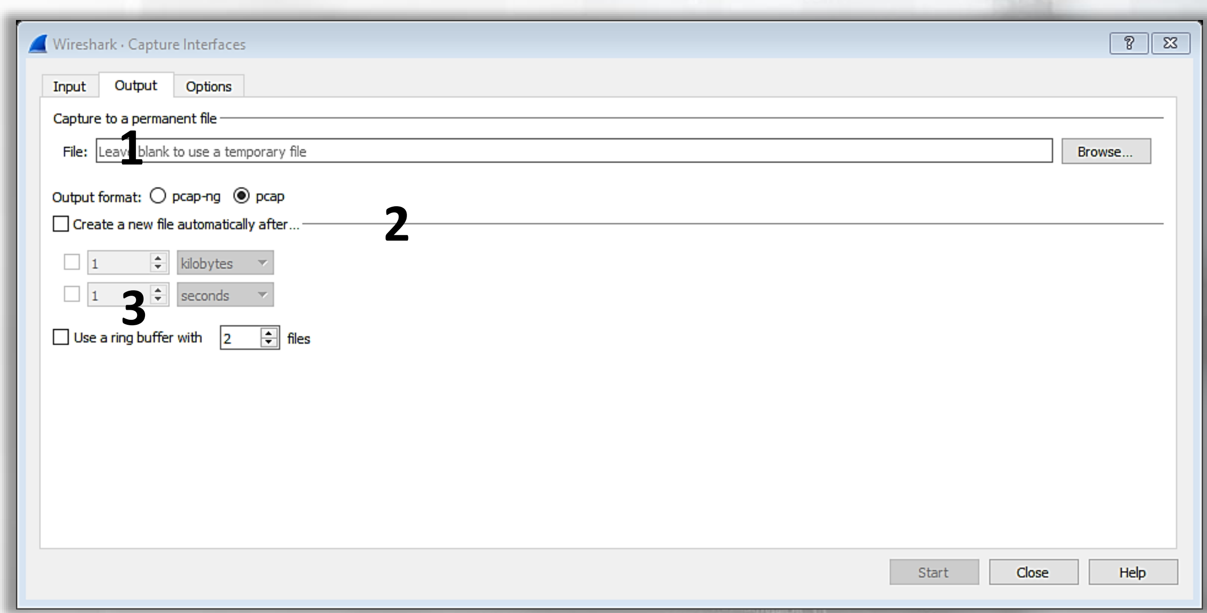
1.3.3 Wireshark

You can choose specific output settings:

1. Name & location for the PCAP file.

2. Output format.

3. Specify when to create a new file, either based on size or time.



Capture to a permanent file

File: Leave blank to use a temporary file

Output format: ○ pcap-ng ● pcap

□ Create a new file automatically after...

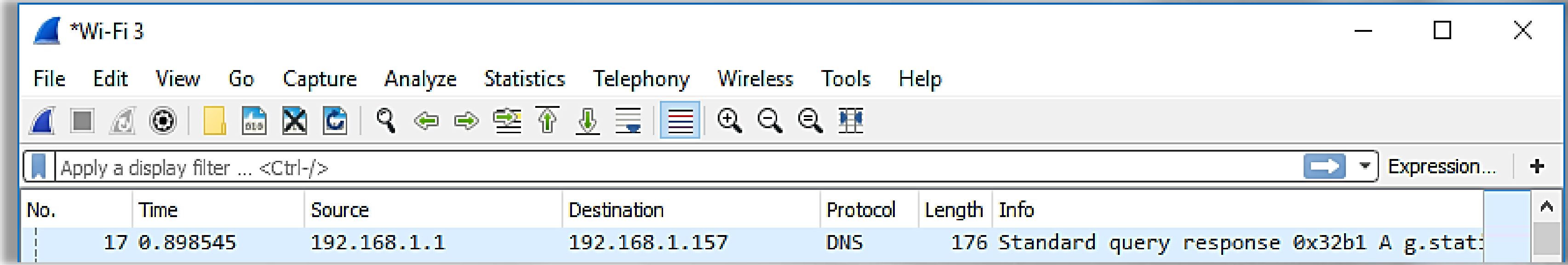
□ 1 kilobytes

□ 1 seconds

□ Use a ring buffer with 2 files

1.3.3 Wireshark

Below is a snippet of the Wireshark window showing 1 packet within the local network. We will be looking at and using Wireshark more in the next module.



No.	Time	Source	Destination	Protocol	Length	Info
17	0.898545	192.168.1.1	192.168.1.157	DNS	176	Standard query response 0x32b1 A g.stat

1.3.3 Wireshark

Wireshark is a great tool for those beginning with network analysis. Although Wireshark has a user-friendly graphical interface, don't let the GUI fool you. It's still a powerful tool for even the most experienced network analysts. You can extend Wireshark's capabilities by writing your own plugins in C or Lua. Wireshark also supports PDML (Packet Description Markup Language), which is used to save network packet dissections.

1.3.4 Dumpcap

It should also be noted that other tools are bundled within the Wireshark distribution, such as **tshark** and **dumpcap**. Both of these tools are command-line based tools and can be used to capture packets.

Dumpcap uses fewer system resources compared to Wireshark.

1.3.5 Tcpdump

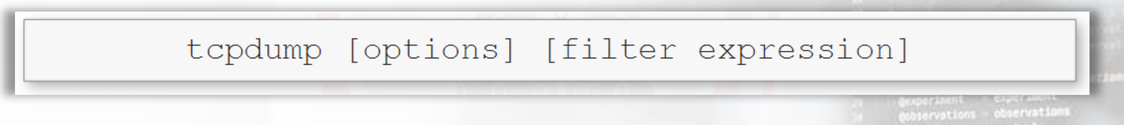
The next tool we'll look at is Tcpdump, which is a powerful packet sniffer that works on most Unix-based systems, such as Linux, FreeBSD, and macOS. It allows the user to intercept and display TCP/IP and other packets being transmitted or received over the network.

Much like Wireshark, tcpdump can filter traffic and save packets to a file for later analysis. Tcpdump offers a lot of options and arguments. The best way to learn more about it is by going through its manual page.

http://www.tcpdump.org/manpages/tcpdump.1.html

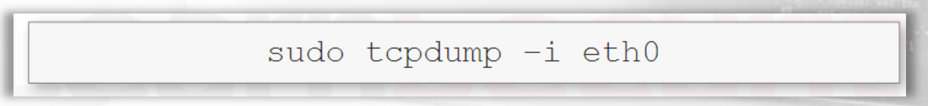
1.3.5 Tcpdump

The first thing we need to know is the syntax that we must use to launch tcpdump:



```
tcpdump [options] [filter expression]
```

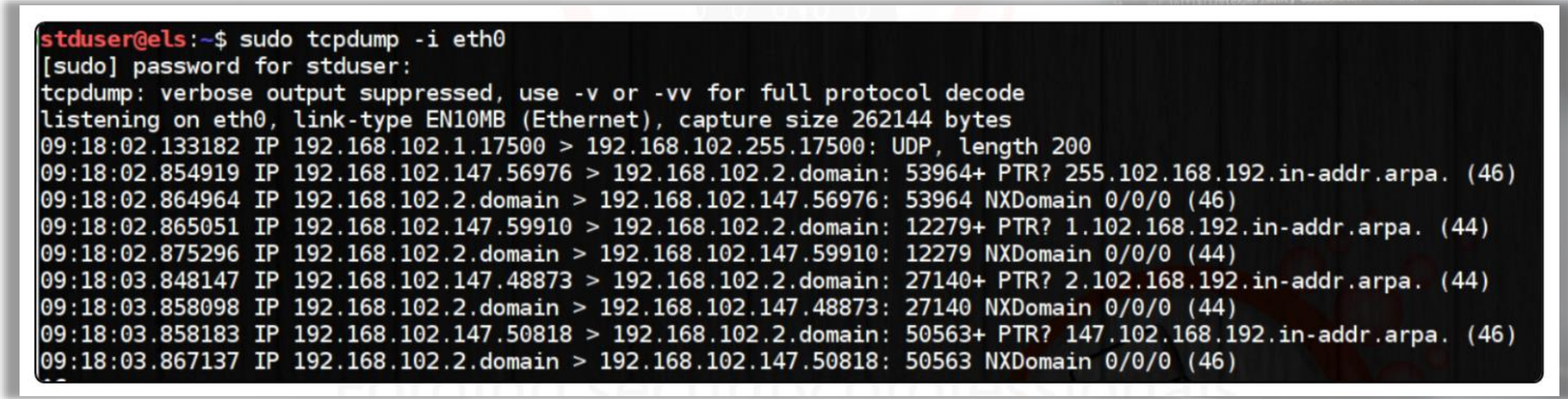
In our example, we want to see all traffic on our main network interface (eth0), so we will use the following command:



```
sudo tcpdump -i eth0
```

1.3.5 Tcpdump

This command states that we want to run tcpdump as root and the “-i” argument indicates the interface to be monitored, eth0 in this case. Since no other options have been added, we will see all packets transmitted.



```
stduser@els:~$ sudo tcpdump -i eth0
[sudo] password for stduser:
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
09:18:02.133182 IP 192.168.102.1.17500 > 192.168.102.255.17500: UDP, length 200
09:18:02.854919 IP 192.168.102.147.56976 > 192.168.102.2.domain: 53964+ PTR? 255.102.168.192.in-addr.arpa. (46)
09:18:02.864964 IP 192.168.102.2.domain > 192.168.102.147.56976: 53964 NXDomain 0/0/0 (46)
09:18:02.865051 IP 192.168.102.147.59910 > 192.168.102.2.domain: 12279+ PTR? 1.102.168.192.in-addr.arpa. (44)
09:18:02.875296 IP 192.168.102.2.domain > 192.168.102.147.59910: 12279 NXDomain 0/0/0 (44)
09:18:03.848147 IP 192.168.102.147.48873 > 192.168.102.2.domain: 27140+ PTR? 2.102.168.192.in-addr.arpa. (44)
09:18:03.858098 IP 192.168.102.2.domain > 192.168.102.147.48873: 27140 NXDomain 0/0/0 (44)
09:18:03.858183 IP 192.168.102.147.50818 > 192.168.102.2.domain: 50563+ PTR? 147.102.168.192.in-addr.arpa. (46)
09:18:03.867137 IP 192.168.102.2.domain > 192.168.102.147.50818: 50563 NXDomain 0/0/0 (46)
```

1.3.6 Berkley Packet Filter

Before we began discussing Wireshark and tcpdump, we looked into libpcap and briefly described what it is. Included within the libpcap library is a filtering language called the **Berkley Packet Filter** (BPF).

With BPF, we can use simple expressions to capture and filter packets. To make complex expressions, we can use nested logic, such as AND/OR statements. **You should become very familiar with BPF**.

1.3.6 Berkley Packet Filter

Throughout the course, we'll use different BPF filters, and it will be pointed out when they are used, but feel free to read up further and get a grasp on BPF as this is a technique you should know.

You can see a list of BPF syntax here. Also, check out these display filter cheat sheets for tcpdump and Wireshark.

http://biot.com/capstats/bpf.html
http://packetlife.net/media/library/12/tcpdump.pdf
http://packetlife.net/media/library/13/Wireshark_Display_Filters.pdf

1.3 Packet Analysis & Tools

In the next module, we'll dive deeper into the topic of packet analysis and look at various PCAP files to identify normal traffic in contrast to malicious traffic using the various tools and techniques discussed in this module.

Module Conclusion

This concludes this module on TCP/IP and Network Primer & Full Packet Logging (including tools). We have covered:

- The concepts and foundation to networking today (OSI & TCP/IP Model, Packet Structure and Encapsulation, and Routers & Switches)
- Full Packet Logging, including tools for analysis

References

References

OSI Model

https://support.microsoft.com/en-us/help/103884/the-osi-model-s-seven-layers-defined-and-functions-explained

IANA

http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml

Tcpdump Man Page

http://www.tcpdump.org/manpages/tcpdump.1.html

Display Filters (tcpdump)

http://packetlife.net/media/library/12/tcpdump.pdf

References

OSI & TCP/IP Model

https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-2000-server/cc958821(v=technet.10)

Wireshark Download

https://www.wireshark.org/download.html

Berkley Packet Filter

http://biot.com/capstats/bpf.html

Display Filters (Wireshark)

http://packetlife.net/media/library/13/Wireshark_Display_Filters.pdf