NTC

Network Traffic Collection

Student Guide

DC3 Cyber Training Academy

The Academy is accredited by the Commission of the Council on Occupational Education (COE).

COE is a national accrediting body dedicated to ensuring quality and integrity in career and technical education.

Product names appearing in this document are for identification purposes only and do not constitute product approval or endorsement by the DC3 Cyber Training Academy or any other entity of the U.S. Government. Trademark and product names or brand names appearing within these pages are the property of their respective owners.

The information contained in this document is intended solely for training purposes and is subject to change without notice. The Academy assumes no liability or responsibility for any errors that may appear in this document.

[v.1901]

CONTENTS

COURSE INTRODUCTION

After completing this course, students will be able to capture selected traffic on a data network and conduct a preliminary analysis of the data.

This course teaches students how to strategically place a monitoring sensor on a network to capture traffic to and from a specific host. Students learn how to evaluate a network, both physically and logically, to determine proper sensor placement. Students also learn how to filter network traffic to comply with wiretap authority, hide the presence of the monitoring workstation on the network, and evaluate captured traffic for the proper content.

LEARNING OUTCOMES *After completing this course, students will be able to:*
Explain basic theory, technologies and components that facilitate network data transmission
Examine network traffic and previously captured data
Perform a logical and physical assessment of a network to identify potential witness devices and the data they contain
Assess a network and identify proper placement of a network-monitoring sensor
Configure network data acquisition tools
Collect and analyze network traffic and system artifacts to identify probing and intrusion techniques

MODULE 1

Introduction to Network Traffic

A basic understanding of how network communications occur is necessary to effectively monitor a network and help investigators separate actions of a suspect from normal network operations.

This module provides an overview of networking technology and identifies common forms of network traffic. This analysis enables students to select the proper logical and physical location on the network to place the monitoring workstation that will gather case information.

OBJECTIVES *After completing this module, students will be able to:*
Identify the differences between the TCP/IP and OSI models
Recognize network traffic
Explain how network connections are established
Identify common network traffic and protocols
Configure and use Wireshark to identify common network traffic and protocols
Use Wireshark to analyze packet header information

Lesson 1

TCP/IP and OSI Models

The TCP/IP network model describes the standard for internet communications. This lesson describes each step in the process of transmitting information from one computer to another across the network using various networking devices, and how these devices relate to the TCP/IP model.

Most browsers mask how information is sent from one computer to another as it traverses the internet. In this lesson, students learn how information is processed in relation to the TCP/IP model.

OBJECTIVES *After completing this lesson, students will be able to:*
Describe the four layers of the TCP/IP model
Name the seven layers of the OSI model
Explain the relationship of the OSI model layers to the TCP/IP model layers

Internet Overview

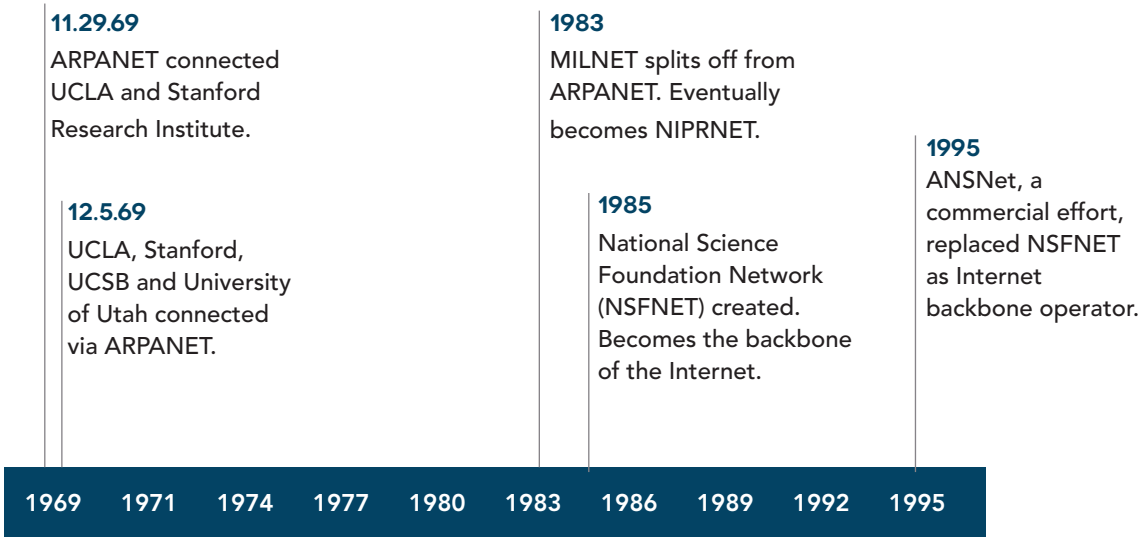
The internet is a massive web of interconnected networks spanning the globe. Over the years, the internet has grown to serve a vast community of government agencies, private organizations and educational institutions. It is the world's largest repository of electronic information.

No one entity owns the internet, and nearly every organization uses it to communicate, share data, conduct research and share resources, such as data management and storage. Individuals also use the internet for personal reasons, such as email and news services.

Internet service providers (ISPs) provide connectivity with the internet to individuals and organizations, generally for a fee. Individuals pay a monthly fee to an International Organization for Standardization (ISO) and may receive software, a username, a password and a broadband connection or a dial-up phone number for access. ISPs also enable large organizations to connect their networks to the internet. Some companies offer dial-up and

DSL services, while other providers such as cable television and telephone companies offer cable modem, DSL and other forms of high-speed access.

KEY DATES IN INTERNET HISTORY



The internet began as ARPANET, the Advanced Research Projects Agency Network, which was implemented in 1969, and initially connected four university research labs.

Over the years, the number of connected hosts grew at a rapid pace. In 1983, the U.S. military's hosts broke off from ARPANET to create MILNET. In 1985, the National Science Foundation Network designed the NSFNET as a government-funded replacement for ARPANET, making it the true internet backbone.

However, in 1995, NSFNET removed itself as the backbone from the internet, causing all internet traffic from that point forward to be routed through commercial connections, which has created the current layout of the internet.

Transmission Control Protocol/Internet Protocol

Transmission Control Protocol/Internet Protocol (TCP/IP) is considered the standard protocol for the internet. TCP/IP can be used for internal networks without internet access, but it must be used for a device to gain internet access.

TCP/IP is a suite of communications protocols governing how data travels between devices and networks throughout the internet.

TCP/IP is a routable protocol that enables computers on different networks to communicate as if they were on the same network. The IP part of this protocol provides the routing capability.

When a message is sent, it is divided into packets. Every client and server in a TCP/IP network has a unique IP address. Each packet carries the IP addresses of both the source and destination computer. TCP/IP determines the travel path between the two computers and then transmits the packets. When a packet reaches its destination, a confirmation is sent to the source computer. This confirmation is why TCP/IP is considered reliable.

During the packets' journey, TCP/IP employs its suite of protocols to enable different types of networks to exchange data.

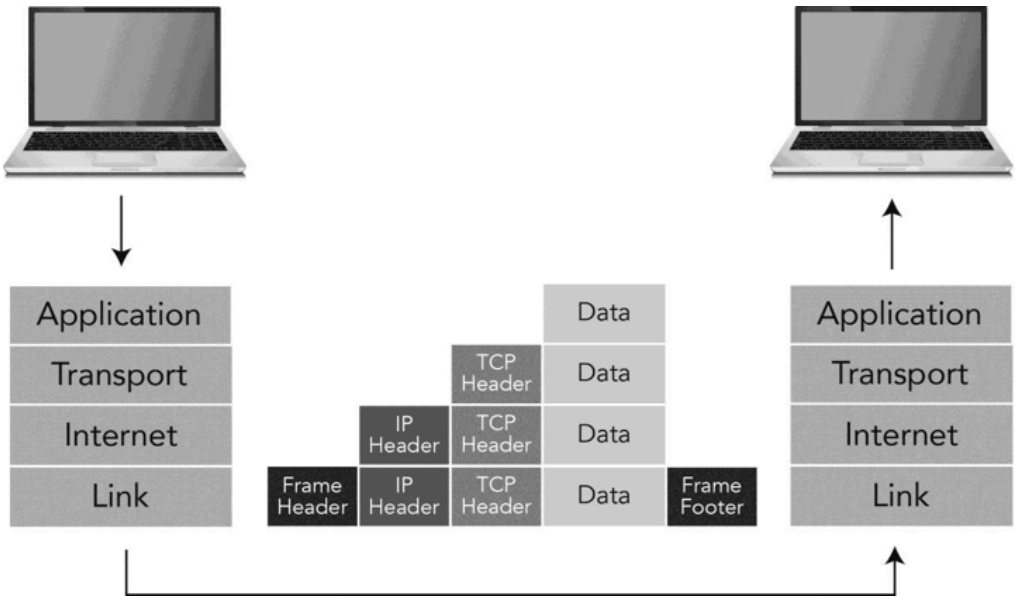
What happens in the actual communications in TCP/IP can be explained through the TCP/IP model, which consists of four layers:

- Application layer
- Transport layer
- Internet layer
- Link layer

Data is transmitted over computer networks in the form of packets contained within frames. A network frame is the TCP/IP model's Layer 1 "container" that encapsulates packetized computer data. An IP packet is data (residing on the hard drive of a computer, for example) that has been divided into smaller chunks of data for transmission. Each smaller chunk is labeled to ensure it reaches the correct destination and can be reassembled into its original form.

A network frame is a two-part entity made up of a header and a payload. The header contains metadata that describes things such as where that packet is going, where it came from, a reference value for error checking and what protocol should be used for the transfer. A packet's payload is the data itself. This standardized structure is read (and often modified by) switching, routing and flow control devices within the network to allow each individual packet to reach its intended destination.

As data moves from one networked computer to another, it traverses the TCP/IP model, receiving various headers and pieces of data added to the leading (and sometimes trailing) edges of it to control and specify how that data is transmitted. This process is called encapsulation.



The TCP/IP model.

TCP/IP Layer Functions Overview

TCP/IP Layer	Description	Protocol and/or Service	Device
Application	Defines how TCP/IP-related applications communicate with the Transport layer services	Email, Web, DNS, HTTP, Telnet, FTP, TFTP, SNMP, SMTP, DHCP, RDP	Gateway, proxy, application firewall, IDS, server
Transport	Provides a method of communication for the source and destination devices	TCP/UDP	Firewall
Internet	Packs the data into IP datagrams and attaches the source and destination addresses	IP, ICMP, ARP	Router, multi-layer switch
Link	Defines how data is physically sent over the network	Ethernet, Serial, Frame Relay, ATM	Bridge, switch, hub, repeater, modem, NIC

TCP/IP Acronyms

ARP	Address Resolution Protocol
ATM	Asynchronous Transfer Mode
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
IDS	intrusion detection system
NIC	network interface card
RDP	Remote Desktop Protocol
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
TFTP	Trivial File Transfer Protocol
UDP	User Datagram Protocol

Application Layer

The Application layer translates data to and from standard text formats, such as American Standard Code for Information Interchange (ASCII) and Unicode. It can also translate that data into another format, such as an image file. The Application layer also handles other data-formatting tasks, such as encryption/decryption and compression/decompression.

The Application layer refers not to the individual applications but to the protocols being used by those applications. For example, a Web browser is an application associated with Hypertext Transfer Protocol (HTTP), or Web, traffic. However, Web browsers also have the ability to handle HTTPS – encrypted Web traffic – in addition to FTP traffic.

Transport Layer

The Transport layer is responsible for transmitting and receiving data across a network, including breaking the data into manageable chunks (segments), applying a sequence number to each segment, and reassembling the segments after transmission. Computers that communicate via TCP/IP may use either a connection-oriented protocol like TCP, or a connectionless protocol like UDP, depending on the type of traffic being transmitted.

TCP has the ability to detect and correct transmission errors, making it a good choice to ensure that data arrives at its destination. For example, to send email using a standard email client, Simple Mail Transfer Protocol (SMTP) is a good choice. SMTP requires a TCP connection so that delivery of the data can be confirmed.

Applications such as video streaming or online gaming most often use UDP because an occasional lost or out-of-order packet does not cause a significant impact.

Internet Layer

The Internet layer deals with network addresses, such as the IP addresses of the classroom computers. IP addresses allow for routing of data between networks. IP addresses themselves can be assigned from a Dynamic Host Configuration Protocol (DHCP) server, or statically assigned by the network administrator or computer user.

Routers

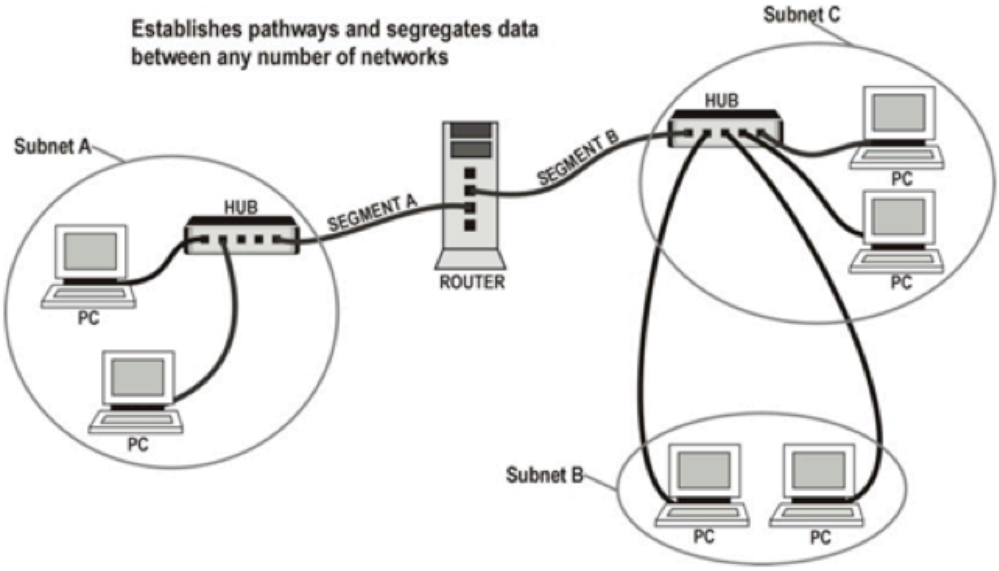
The most common devices associated with the Internet layer are routers, which link separate networks or local area network (LAN) segments and establish pathways for data packet transmissions.

Each network has an IP network address. Routers use this address to transmit packets to the correct destination. Routers also:

- Transmit data packets across different types of networks
- Fragment data packets to fit different frame sizes of various networks
- Can be configured to segregate secure data and prevent them from being sent to specified networks
- Collect and assemble information from remote routers about network routes; this information is used to identify reliable pathways
- Do not broadcast data packets

Each port on a router is in essence a separate network interface card (NIC) with a unique media access control (MAC) address. Therefore, as packets move from one router to another, the MAC address changes from router to router. The original source and destination IP addresses remain the same, regardless of how many routers a packet encounters.

A router model.



Routers on networks exchange information about paths to computers attached to them through a process known as convergence. Convergence information is stored in routing tables, which contain the network portion of the host computer's IP address.

Routing assumes that addresses convey at least partial information about where a host is located. This permits routers to forward packets without having to rely on a complete listing of all possible destinations.

Routing involves two basic activities:

- Path determination
- Switching

Path determination enables a routing protocol to determine the best direction to route a packet. It is complex because the determination differs based on the routing protocol used.

Switching involves the router forwarding the packets independently through the network. The router forwards the packets based on the IP address. If the IP address is not in the router's routing table, the router drops the packet.

Link Layer

The Link layer is responsible for examining MAC addresses. The MAC address is the primary identifying stamp on all data packets transmitted to or from a computer within a LAN. Once the data packet from a machine

leaves the LAN for another network, the MAC address is changed to reflect the last machine retransmitting the packet.

The MAC address is a 48-bit binary number, which is represented in a 12-digit hexadecimal format so that it can be read easily. It may be presented as six groups of two hex digits, as in the following example:

00:02:2D:87:BE:8D

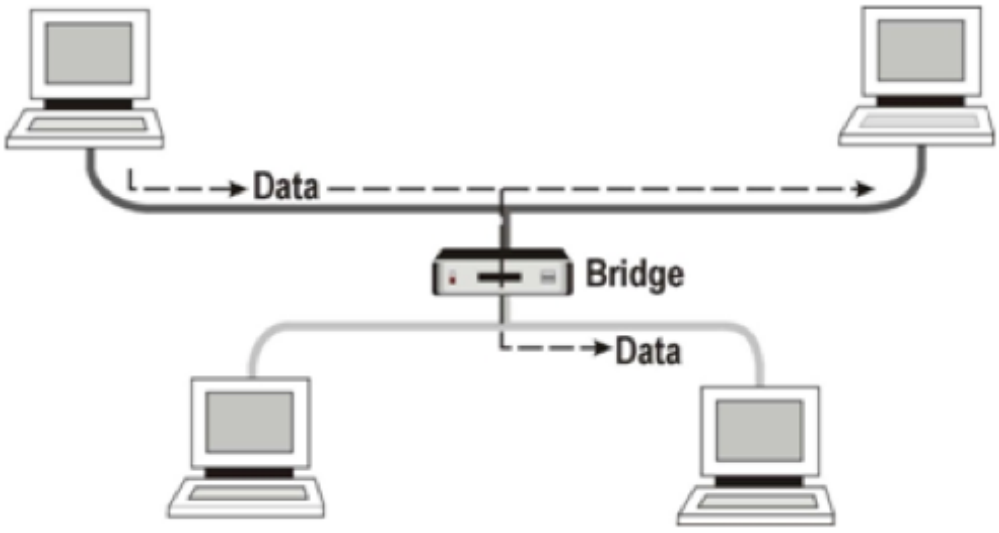
Each MAC address can be split in half when read. The first half, or the first three bytes (24 bits), is called the organizationally unique identifier (OUI) and is unique to a manufacturer. The second half is unique to a specific card.

To find the manufacturer of the NIC, visit: http://standards.ieee.org/regauth/oui/index.shtml

Most networking devices fall into the Link layer. The following sections cover these devices.

Bridges

A bridge is a unit that joins two separate segments of the same network. It can also can be used to divide an overloaded network by creating separate broadcast, or collision, domains. A bridge can also connect two networks that are dissimilar.



An example of a bridge.

A bridge decides whether data packets should be sent from one collision domain, across the bridge, and into the second collision domain, based on the MAC address of the sending and receiving nodes. If the sending and receiving nodes are on the same segment, the bridge ignores, or drops, the packets.

Switches

Switches are networking devices that meet the demand for faster connections and more bandwidth. Although switches resemble hubs, they help increase the speed of the network by providing dedicated bandwidth to each port. In contrast, hubs share the bandwidth among all ports.

A switch functions like a cross between a bridge and a hub. Switches cut the amount of broadcast traffic on the network segment by switching network packets from the incoming port and sending them directly to the receiving computer's port.

By decreasing the amount of broadcasts on the network, the number of collisions on the network segments is also lowered, improving overall performance. Like intelligent hubs, switches can be managed, allowing individual port configuration and monitoring from across the network.

Switches direct data packets between ports based on MAC addresses. Switches have the ability to broadcast to all ports when necessary, but differ from hubs in that they can limit traffic to the sender and receiver ports without broadcasting.

NICs

A network interface card (NIC) is an adapter in a computer that enables the computer to connect to a network. Each NIC is made for the network type it supports, such as Ethernet. Some cards are formatted as separate expansion cards, and others are integrated into the motherboard.

When a computer makes a request to communicate with the network, the OS sends the request to the NIC. The NIC converts the request into the proper type of data packets to be transmitted over the network. It then monitors network traffic flow and sends the packets at the appropriate time when there is an opening.

In addition to preparing and sending packets, the NIC checks the MAC addresses of passing network transmissions. If they are addressed to the computer, the NIC then copies the packet for the computer.

NICs decide whether to read incoming data packets based on the MAC address.

Hubs

A hub is used to join several nodes together at a single site. Its main functions are to connect nodes, organize cabling, and transmit signals to anything that is attached to it, including other segments of the network. Hubs generally broadcast data packets to every node on the hub, but differ among two types:

- **Passive broadcast hub:** Performs no signal regeneration.
- **Active broadcast hub:** Enhances signal transmission by regenerating signals and filtering noise.

An intelligent hub is essentially an active broadcast hub that contains network management functions used to gather information on network traffic and error detection. Most intelligent hubs allow a user to monitor individual ports and close a port if problems arise.

Hubs do not decide when or where to send data packets. They simply broadcast the data to all ports.

Repeaters

To boost signals, analog repeaters amplify the signal and digital repeaters regenerate the signal. These devices can relay signals between networks that use different types of protocols or cabling.

Repeaters do not decide when to send data. They simply receive data packets in one port, regenerate or amplify them, and send them to the other port.

Modems

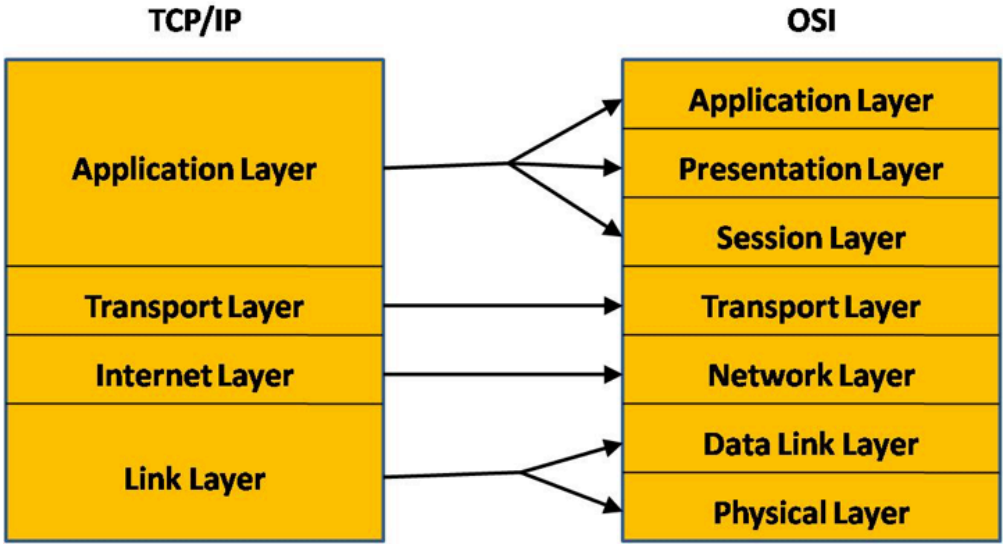
Modems handle communication transmitted over telephone lines between computer systems. Most modems have fax capabilities.

PCs are digital devices, and the telephone system is analog. The modem is the component that converts or modulates the PC's digital code to analog so it can be sent over phone cables. Likewise, when receiving information, the modem converts or demodulates analog signals to digital code before transmitting data to the PC.

TCP/IP Model vs. OSI Model

When comparing the TCP/IP model with the OSI seven-layer model, OSI is considered a conceptual model of how communications should flow from one network to another. It provides a standard for other protocols to use.

TCP/IP represents the actual implementation of how internetwork communications occur. In the TCP/IP model, the Application layer absorbs the functions of OSI's Application, Presentation and Session layers. The functions of OSI's Data Link and Physical layers are also combined to represent the TCP/IP Link layer.



A comparison of the layers in the TCP/IP and OSI models.

Lesson 2

Understanding Network Traffic

A basic understanding of network communications helps investigators separate suspect actions from normal network operations and to employ appropriate capture and display filters.

This lesson introduces common forms of network traffic and explains how the different network layers affect network monitoring. This lesson focuses on the TCP/IP (or DoD) network model.

OBJECTIVES *After completing this lesson, students will be able to:*
Explain network and physical network addressing
Identify common network traffic and protocols

MAC and IP Addresses

When setting up a network monitoring device, MAC or IP address filters can be used to ensure that the device captures only data pertaining to the investigation. It is important to know when the use of MAC or IP address filters is beneficial. The following explanations assume investigators are dealing with common Ethernet-based networks.

TCP/IP Model

The TCP/IP model can be used to illustrate how IP addresses relate to MAC addresses. The model shows how each layer builds on the previous one. Inside the packet is a header for each protocol used on each layer, and only one protocol can sit on each layer. For example, when a Web browser communicates with a Web server, each packet has:

- An Ethernet header with MAC addresses (physical computer addresses)
- An IP header with IP addresses (logical computer addresses)
- A TCP header with ports (associates packets with applications)
- An HTTP layer with HTTP data

	Layer	Protocol	Address Found in Header*
4	Application	HTTP, SMTP, FTP, DNS, Telnet ...	Data
3	Transport	TCP, UDP, ICMP, IGMP	Ports
2	Internet*	IP, IPSEC	IP addresses
1	Link**	Ethernet, Token ring, Frame relay ...	MAC addresses

* Contains a source and destination address

** The Link layer is also commonly called the Network Interface layer or Network Access layer.

Addressing Schemes

Addressing and identifying devices on a network correctly is an important function for all networks. With most networks, this is accomplished using two basic addressing schemes:

- Physical address (MAC)
- Logical address (IP)

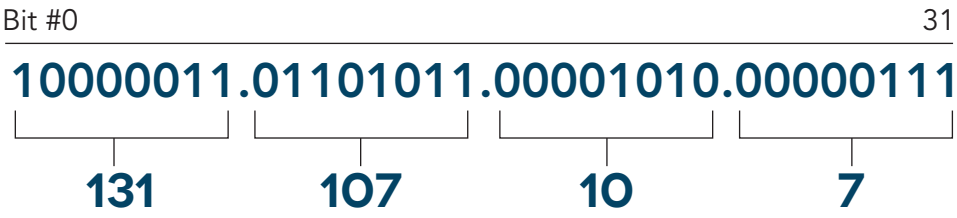
The physical address is typically used for communication between devices in the same network segment. The logical address is used for communication between different network segments. For Ethernet networks, the MAC address is used for physical addressing. IP addresses are typically used for logically addressing packets destined for other network segments.

Workstations can have either a permanent IP address or one that is assigned during each network connection. For clients on an isolated LAN, the administrator can assign unique IP addresses. However, to communicate with the internet, a user must have a routable IP address to avoid duplicates and permit routing of packets to their intended destination

IP Addresses

An IP address (also called IPv4, or IP version 4) is a 32-bit numeric address written as four sets of numbers, called octets, separated by periods. Although actually stored as a series of eight binary digits (or bits), when represented using a decimal-based numbering system, the value of each octet can range from 0 to 255. A valid IP address cannot consist of all 0s or all 1s.

TYPICAL CLASS B INTERNET ADDRESS 131.107.10.7



Binary IP Addressing

Computers at their lowest level only understand the 1s and 0s of the binary number system, or base-2. The binary number system represents all values as a series of binary digits consisting of 0s and 1s. IP addresses are typically represented as a set of four decimal numbers separated by periods. Therefore, these typical IP addresses must be translated by the computer from their decimal representation to a binary form to be understood.

Because each octet in an address is limited to eight bits, the position of each of the eight bits represents a specific value for that bit, starting at 1 for the far right bit and progressing to 128 for the far left bit. This allows a total value range of 0 through 255. The following chart illustrates the use of base-2 in converting the decimal 131 to its binary number equivalent of 10000011.

Binary Conversion of Decimal 131								
Position	7	6	5	4	3	2	1	0
Base-2	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Decimal Value	128	64	32	16	8	4	2	1
Binary No. is 10000011	1	0	0	0	0	0	1	1

Classes of IP Addresses

IP addresses are divided into several class types. Classes A, B and C are used for government and commercial addresses. Classes D and E are reserved for multicasting, which is the transmission of data to many recipients simultaneously.

Each class allows for a specific maximum number of subnets and end nodes. This table describes the characteristics of each IP address class type:

Class	Leading Bits	Start	End	Number of Networks	Hosts per Network
A	0	0.0.0.0	127.255.255.255	128	16,777,214
B	10	128.0.0.0	191.255.255.255	16,384	65,534
C	110	192.0.0.0	223.255.255.255	2,097,152	254
D (multicast)	1110	224.0.0.0	239.255.255.255	Not defined	Not defined
E (reserved)	1111	240.0.0.0	255.255.255.255	Not defined	Not defined

Reserved IP Addresses

This table identifies IP addresses/ranges that are reserved for specific functions:

Description	IPv4 Address Range
Reserved for unroutable networks	• 10.0.0.0 to 10.255.255.255 • 172.16.0.0 to 172.31.255.255 • 192.168.0.0 to 192.168.255.255
Automatic Private IP Addressing (APIPA)	• 169.254.0.0 to 169.254.255.255
Reserved for loopback NIC testing	127.0.0.1
Reserved for routing tables; refers to entire network	128.5.0.0
IDs an entire network	X.X.X.0
Broadcast	X.X.X.255

Classless Inter-domain Routing

The class system provides a finite number of IPv4 addresses. Consequently, the number of unassigned internet addresses is running out. A scheme called Classless Inter-domain Routing (CIDR) has been introduced as a replacement for the system based on classes A, B and C.

With CIDR, IP addresses are assigned in blocks. A single IP address can be used to identify many unique IP addresses. A CIDR IP address looks like a normal IP address except that it is appended with a slash (/) followed by a number. This end number is called the IP (or network) prefix. An example of a CIDR address is 162.200.0.0/12.

The IP prefix designates how many bits define the network portion of the address, with the remaining bits identifying the hosts. In the previous example, 162.200.0.0/12, the first 12 bits of the address identify the network, and the remaining 20 bits are used to identify the host.

CIDR addresses also reduce the size of routing tables and allow for more IP addresses for subnetting and supernetting within organizations.

Static IP Address

A static Internet Protocol (IP) address (static IP address) is a permanent number assigned to a computer by an Internet service provider (ISP). Static IP addresses are useful for gaming, website hosting or Voice over Internet Protocol (VoIP) services. Speed and reliability are key advantages. The opposite of a never-changing static IP address is an ever-changing dynamic IP address. A dynamic IP address is just a regular address like a static IP is, but it's not permanently tied to any particular device. Instead, they are used for a specific amount of time and then returned to an address pool so that other devices can use them via the Dynamic Host Configuration Protocol (DHCP). Dynamic addresses provide a way for IP addresses to be reused when they're not in use elsewhere, providing Internet access for many more devices than what would otherwise be possible.

Dynamic Host Configuration Protocol

Network administrators may use DHCP to assign dynamic IP addresses to individual devices on a network. Addresses are assigned from a pool of preregistered addresses. DHCP saves time by eliminating the steps to assign IP addresses manually to new network equipment. It also tracks all assigned addresses automatically. With DHCP, a computer or other device may be assigned a different IP address every time it accesses the network. In some cases, a device can change its IP address between logon and logoff. ISPs frequently use DHCP for their dial-up users and even for higher-speed broadband connections.

IPv6

IPv6 was developed by the Internet Engineering Task Force (IETF) to deal with the long-anticipated problem of IPv4 address exhaustion. IPv6 is intended to replace IPv4. While the adoption of IPv6 standard goes much slower than initially planned, the transition is taking place.

Created in 1998, Internet Protocol Version 6 (IPv6) is a specification that defines a new method for communications addressing on packet-switched networks. Although it has yet to be adopted for widespread use, IPv6 is an accepted standard that is being implemented on newer computers and networking equipment. This standard provides a much larger address space than the IPv4 standard and uses 128-bit addresses.

An IPv6 address is written as eight groups of hexadecimal digits as seen in the following example:

2001:0db8:85a3:08d3:1319:8a2e:0370:7334

Because the IPv4 address space is a recognized subset of IPv6, both protocols can coexist on the same physical network. Due to the size of the IPv6 address space (3.4×10^{38} unique addresses), it is expected to provide greater flexibility and eliminate the need for NAT. Occurrences of IPv6 addressing should continue to increase in the future, eventually replacing IPv4.

IPv6 is enabled by default in more recent versions of Microsoft Windows, including Windows 7, Windows 10, and Windows Server 2016. Mac OS X and newer versions of Linux also come bundled with IPv6.

IPv4 and IPv6

Regarding IP network packet structure, IPv4 and IPv6 are in current use and markedly different. The following table addresses a few of these major differences.

Category	IPv4	IPv6
Header size	160 bits (20 bytes)	320 bits (40 bytes)
Packet size	Up to 65,535 bytes (Link layer limit)	Up to 65,535 bytes (Link layer limit) (IPv6 jumbogram mode allows packet size to reach 4,294,967,295 bytes via a 32-bit packet length field)
Addressing	32-bit address space 2*32 unique addresses (4,294,967,296) Dotted decimal notation representing four eight-bit fields, e.g., 192.149.252.76 Each interface must have a single, unique IP address	128 bit address space 2*128 unique addresses (340,282,366,920,938,463,463,374,607,431,768,211,456) Hexadecimal notation representing 16 eight-bit fields, e.g., 3FFE:F200:0234:AB00:0123:4567:8901:ABCD, or FE80::4CD9:A8E5:154D:F954 Each interface may have several unique IP addresses
Configuration	DHCPv4 Manual	DHCPv6 Router Advertisement (RA) Stateless Address Autoconfiguration (SLAAC) IPv6 is self/autoconfiguring
Address resolution	ARP	Multicast Neighbor Solicitation
Routing	Public and private IP address ranges	Public and temporary IP addresses; however, temporary addresses are also publicly routable

MAC Address

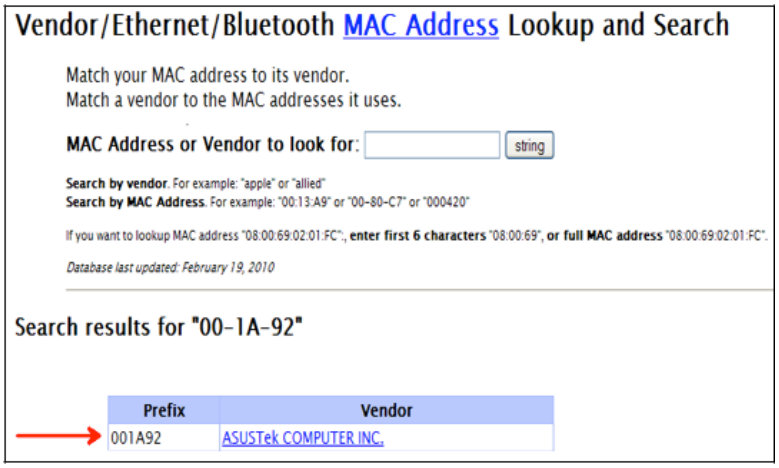
A MAC address is used by the Ethernet protocol. Ethernet runs on Layer 2 of the OSI model and units of data transmitted at this layer are known as frames. Layer 2 is responsible for getting all frames of data on a LAN to the next hop or destination on the same LAN. MAC addresses can only be used to send a frame between devices on a LAN and cannot get a frame to another network (i.e., MAC addresses cannot route packets through a router). MAC addresses are found on the NIC and are rarely changed. A MAC address consists of 6 bytes and is usually written as 12 hexadecimal digits separated by colons or dashes as seen in the following example:

00:00:1a:34:32:93

NOTE

To determine the vendor of a MAC address, visit: http://www.coffer.com/mac_find/

A MAC address is 48 bits. The first six characters, or 24 bits, of the MAC address are the OUI, which is 3 bytes and identifies the vendor or manufacturer of a specific NIC.



When a user types a MAC address into the lookup box, coffer.com will return the vendor based on its database of OUI identifiers.

Wireshark, a free protocol analyzer, can identify the vendor by examining the OUI of each MAC address within a given capture file. With Wireshark installed, a file containing the list of vendors and the corresponding OUIs is located in C:\Program Files\Wireshark\manuf. This file with no extension is a text file and can be opened with WordPad. A user can add or delete entries from the list by editing the text file.

Although an OUI can help identify a computer's vendor, keep in mind that a MAC address can be spoofed using tools such as macshift, macchanger, macspoof and ifconfig. A spoofed MAC address prevents someone from identifying the hardware vendor of the computer's NIC (and possibly the system).

macchanger

The macchanger command-line tool allows a user to view and change the user's current MAC address in Linux. In Linux, the ifconfig command can also be used to change a MAC address. The interface must be brought down first when using either ifconfig or macchanger to change a MAC address.

```
[root@localhost root]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0C:29:92:C8:E8
          inet addr:192.168.242.25  Bcast:192.168.242.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4 errors:0 dropped:0 overruns:0 frame:0
          TX packets:56 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:631 (631.0 b)  TX bytes:4351 (4.2 Kb)
          Interrupt:5 Base address:0x2000

[root@localhost root]# ifconfig eth0 down
[root@localhost root]# ifconfig eth0 hw ether 00:0C:29:12:34:56
[root@localhost root]# ifconfig eth0 up
[root@localhost root]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0C:29:12:34:56
          inet addr:192.168.242.25  Bcast:192.168.242.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4 errors:0 dropped:0 overruns:0 frame:0
          TX packets:56 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:631 (631.0 b)  TX bytes:4351 (4.2 Kb)
          Interrupt:5 Base address:0x2000
```

Macchanger provides a variety of features such as changing the address to match a certain vendor or completely randomizing it.

Broadcast MAC/IP Addresses

When a computer needs to address a transmission to all other computers on the same LAN segment, it uses the broadcast MAC or IP address, which is:

- **MAC:** FF:FF:FF:FF:FF:FF
- **IPv4:** x.x.255.255 (where x.x is the local network identifier)

This broadcast IP address is an example for a Class B network. The x.x represents the number of bits in the local network identifier. If this were not a Class B network, the number of bits would be different and would reflect the actual local network identifier.

When a computer wants to communicate with every computer on the LAN, a broadcast packet is used with protocols like ARP. Broadcast traffic can be confusing when investigators are trying to target the location of a computer on a network. Investigators must know that even though there is broadcast traffic with the target's MAC or IP address, it does not mean that the sniffer is in the right location. Broadcast traffic is sent to every device on the LAN regardless of location.

How Does This Affect Network Monitoring?

Protocol/Device	Effect on MAC/IP Address Filters
DHCP	New IP address could be allocated to the target/victim computer.
NAT/PAT	Changes IP address information in packet.
Virtual machines	MAC and IP address information is usually virtualized within the host OS and not seen on the wire.
Ethernet	MAC address information is changed when packet leaves LAN. MAC address on NIC rarely changes.
IP	IP address can be used outside of LAN. IP address can be easily changed.

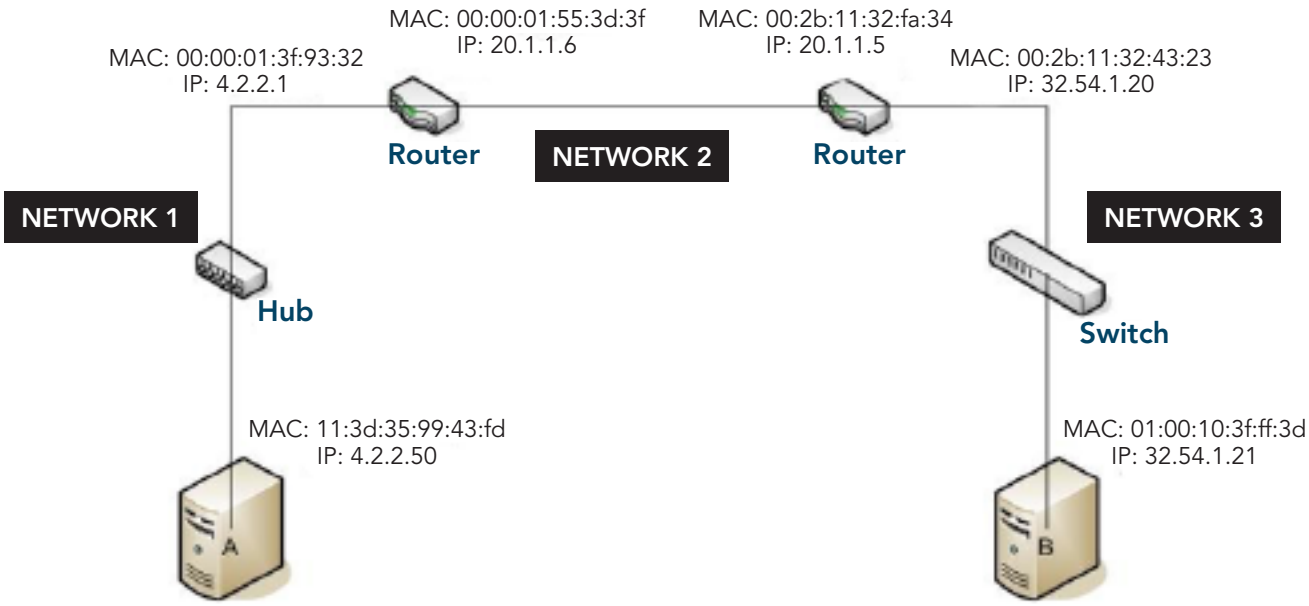
When using a MAC address as a filter, the monitoring station must be on the same network as the NIC using that MAC address. When the packet travels to another network, all MAC address information within the packet is lost.

The IP address information inside a packet will stay constant as it traverses different networks (with a few exceptions like NAT and PAT). This means the same IP address can be used effectively as a filter on multiple networks.

If using an IP address means the network monitor is not restricted to the same network as the target, why not use an IP address as a filter every time? This chart shows situations that affect MAC and IP address filters.

Exercise: MAC/IP Address Effects

This exercise demonstrates how MAC and IP address information is affected as a packet traverses different networks.



Using the diagram, fill out how the information for a single packet sent from Computer A to Computer B will change as it traverses the different networks.

Network 1

Source MAC address: _____

Destination MAC address: _____

Source IP address: _____

Destination IP address: _____

Network 2

Source MAC address: _____

Destination MAC address: _____

Source IP address: _____

Destination IP address: _____

Network 3

Source MAC address: _____

Destination MAC address: _____

Source IP address: _____

Destination IP address: _____

NAT and PAT

When choosing an IP address as a filter, investigators must know the effects of Network Address Translation (NAT) and Port Address Translation (PAT). Both of these technologies will change the IP address information within a packet. This is usually done at a firewall and will affect the placement of a sniffer and the configuration of the filter.

IPv4 Problem

The current, 32-bit IPv4 IP address schema has only about 4 billion unique combinations (2^{32}). Devices that want to communicate on the internet must use a unique IP address. In the internet's infancy, 4 billion addresses were plenty, but today the number of internet users and devices has reached hundreds of billions, and there are not enough IPv4 addresses. To address this issue, IPv6 was developed as a new schema that adds trillions of new addresses. Though IPv6 solves that problem, it creates other issues. Implementing IPv6 will require a costly and time-consuming upgrade to the internet's infrastructure. In the meantime, NAT and PAT provide for the growth of the internet by allowing computers a method to share IP addresses.

Private Addresses

Private addresses are IP addresses that are reserved for private networks and are not routable on the internet. Private addresses allow system administrators to configure IP information on a local network without having to worry about the local IP addresses conflicting with internet (or public) IP addresses. These local private addresses use NAT and PAT to communicate on the internet, and are represented in the following examples:

- 10.x.x.x
- 172.16.x.x through 172.32.x.x
- 192.168.x.x

NAT

NAT usually occurs at the gateway firewall for the network hosting the computer using the private IP address. It is a translation of a private IP address to a public IP address. The firewall will maintain a table that matches private and public IP address translations.

Translations can occur in two ways: through the use of Static NAT or Dynamic NAT.

Static NAT maps internal (private) network IP addresses to registered (public) IP addresses on a one-to-one basis. An internal, unregistered address will always map to the same registered address.

- 10.10.10.1→Router→20.2.28.1
- 10.10.10.2→Router→20.2.28.2
- 10.10.10.3→Router→20.2.28.3

Dynamic NAT also maps internal IP addresses to registered IP addresses on a one-to-one basis. However, Dynamic NAT assigns public addresses from a pool of addresses registered to the organization. Address assignment could differ for each request depending on what addresses are available at the time the mapping translation takes place. The address is returned to the pool after its use is complete and reassigned as new requests are received. Machines may or may not be assigned the same IP address with each request depending on address availability.

- 10.10.10.1→Router→20.2.28.5
- 10.10.10.2→Router→20.2.28.17
- 10.10.10.3→Router→20.2.28.11

PAT

PAT usually occurs at the border between the public network and the private network, and replaces the source port in addition to the source IP address. PAT gives all private internal IP addresses the ability to share a single public, registered IP address to communicate outside the LAN and with the internet. The public address is the IP address used by the NIC facing the internet. PAT is typically performed by a border router or a firewall.

The source IP (private address) and the source port uniquely identify the computer and the application transmitting the packet. Since there is only one public IP address, PAT must also change the source port to maintain the uniqueness of the IP address/port combination. PAT selects a unique outgoing port to put in the outgoing packet, and keeps a translation table

between outgoing port and source IP/port combination. When a packet returns with the public IP address and the unique port, PAT inserts the proper private address and port to allow the correct application to receive the returned packet.

How Does This Affect Network Monitoring?

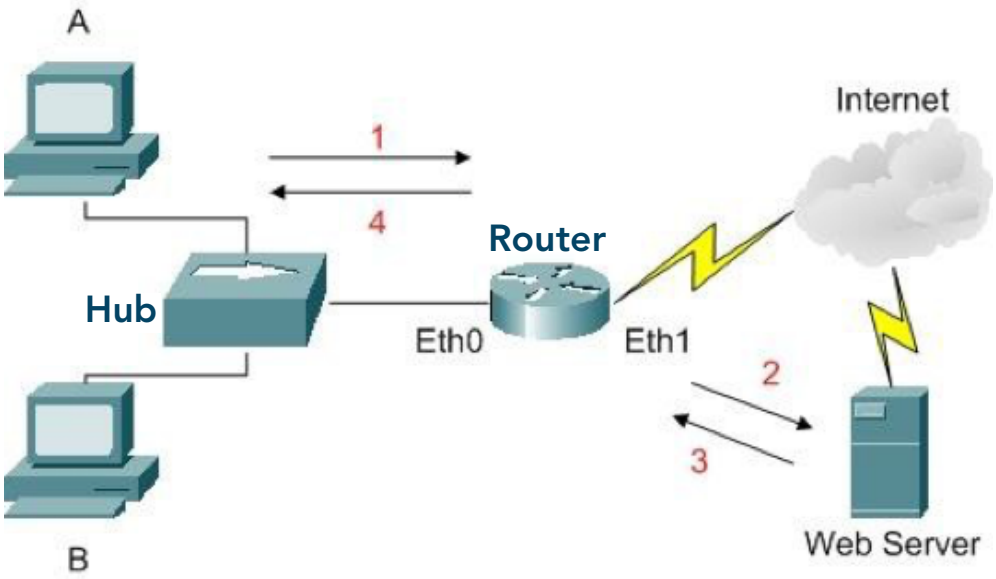
An IP address is a common filter used on a network monitor. Knowledge of how IP addresses in a packet are affected by NAT and PAT is essential to network monitoring. A private IP address would not be a valid filter outside the firewall or NAT/PAT device.

Also, the public addresses used by NAT/PAT are not valid filters inside the firewall.

PAT Example

In this example, the router performs PAT for workstations A and B when they communicate with hosts on the internet. Here are the IP addresses of the devices:

- **Workstation A:** 192.168.0.2
- **Workstation B:** 192.168.0.3
- **Router interface Eth0:** 192.168.0.1
- **Router interface Eth1:** 212.64.119.72
- **Web server:** 65.207.86.178



When Workstation A attempts to view a web page on the remote Web server, this process takes place:

Step	Action
1	Workstation A sends a connection request to the remote Web server. The request has these attributes: • **Source IP:** 192.168.0.2 • **Destination IP:** 65.207.86.178 • **Source port:** 1025 • **Destination port:** 80
2	The router receives the request and forwards it to the remote Web server but first performs PAT. PAT changes the source IP address and uses a new source port assigned by the router to manage the to manage the communication. Afterward: • **Source IP:** 212.64.119.72 • **Destination IP:** 65.207.86.178 • **Source port:** 35001 • **Destination port:** 80
3	After receiving the request, the Web server replies to the IP address of the router's external interface. The reply from the Web server includes the following information: • **Source IP:** 65.207.86.178 • **Destination IP:** 212.64.119.72 • **Source port:** 80 • **Destination port:** 35001
4	The router receives the reply from the Web server and uses PAT to translate the reply and send it to the workstation that originated the request. The port 35001 is used to internally reference the workstation and source port that was originally used. • **Source IP:** 65.207.86.178 • **Destination IP:** 192.168.0.2 • **Source port:** 80 • **Destination port:** 1025

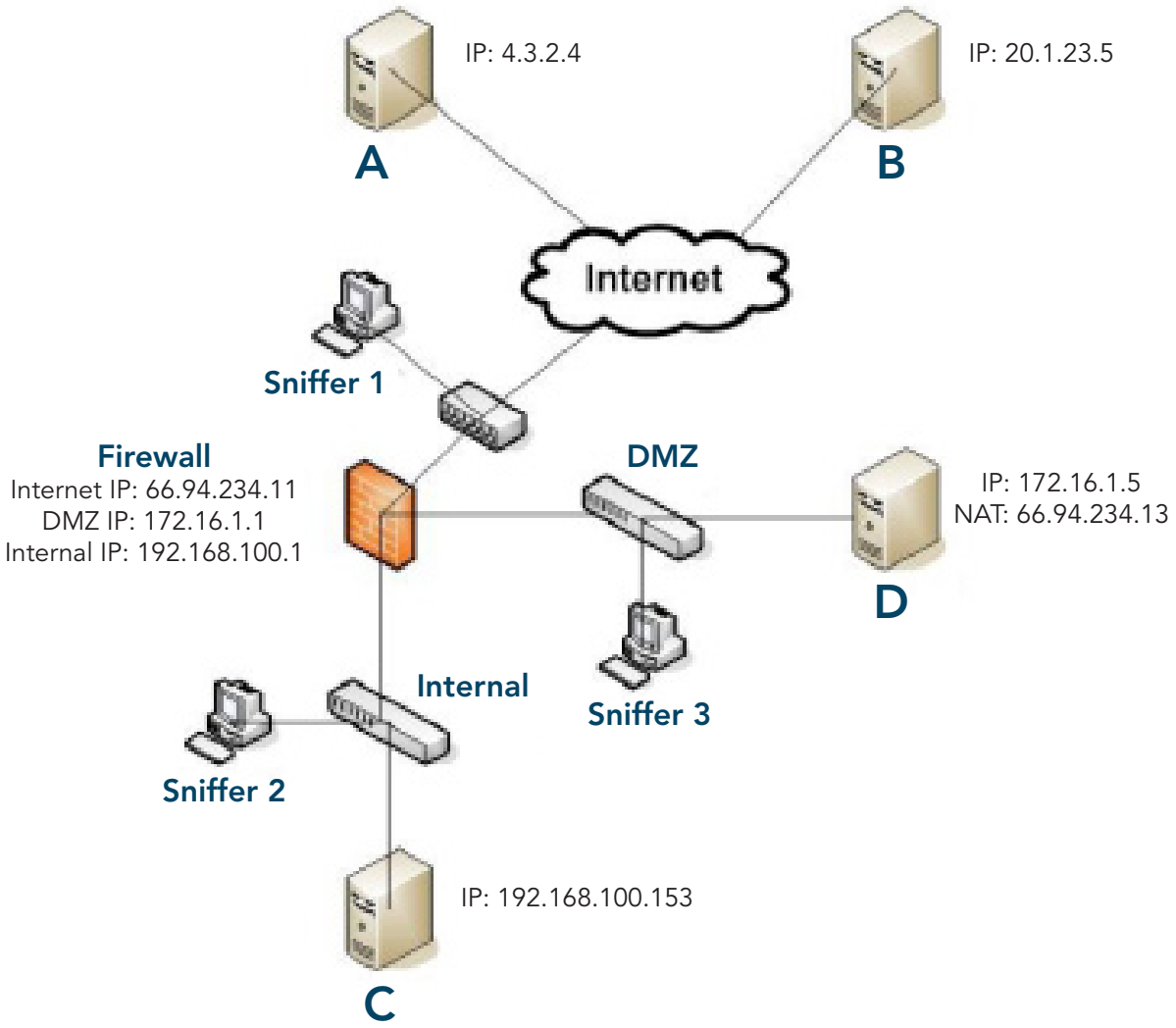
When a device uses PAT to translate a request, it records the following information, which enables it to keep track of the translations it performs for many hosts at the same time:

- Original source IP
- Destination IP
- Source port
- Destination port
- TCP sequence numbers

The fact that a device can change the source IP address of a network transmission is important. It illustrates that the source IP address obtained from a sniffer may not be the IP address of the computer that originated the request, but of the device that last performed PAT. All transmissions leaving a network may have the IP address of the device performing PAT, even though many computers may be behind that device.

Exercise: NAT Scenarios

This exercise demonstrates how IP address information is affected as a packet traverses to and from public to private networks. Using the diagram, fill out how the IP information in a packet will change depending on what sniffer information is being viewed.



NAT Scenario 1

Sniffer 2: Packet from Computer C to Computer B

Source IP address: _____

Destination IP address: _____

Source port: 2400

Destination port: 80

Sniffer 1: Packet from Computer C to Computer B

Source IP address: _____

Destination IP address: _____

Source port: Does it change? Y/N

Destination port: Does it change? Y/N

NAT Scenario 2

Sniffer 1: Packet from Computer A to Computer D

Source IP address: _____

Destination IP address: _____

Source port: 32000

Destination port: 443

Sniffer 3: Packet from Computer A to Computer D

Source IP address: _____

Destination IP address: _____

Source port: Does it change? Y/N

Destination port: Does it change? Y/N

Lesson 3

Common Network Protocols

Network protocols are rules and procedures for communication between various devices. Protocols function at different layers within the TCP/IP and OSI models. This lesson introduces common network protocols and their functionality.

OBJECTIVES
After completing this lesson, students will be able to:

- Explain how network connections are established
- Identify common network traffic and protocols

TCP

The Transport layer uses TCP, a connection-oriented communication protocol for services that require high reliability. TCP uses a process called the three-way handshake to establish communications between a client and a server. These are defined as:

- **Client:** A host seeking to use the resources of a server
- **Server:** A host that receives requests to use its resources

Three-way Handshake

A TCP connection is a fully reliable, connection-oriented communication protocol that guarantees data delivery. Before any TCP data is exchanged between a client and a server, the three-way handshake establishes the connection. This marks the beginning of a TCP session.

TCP Flags

TCP uses eight flags to help manage a connection-oriented communication. Each flag consists of one bit in the TCP header with two states, on or off. These four flags most affect network monitoring:

Flag	Description
SYN	Synchronize or begin a communication
ACK	Acknowledgment or acknowledge that a packet was received
RST	Reset or end the communication immediately
FIN	Finish or end the communication (nicely)

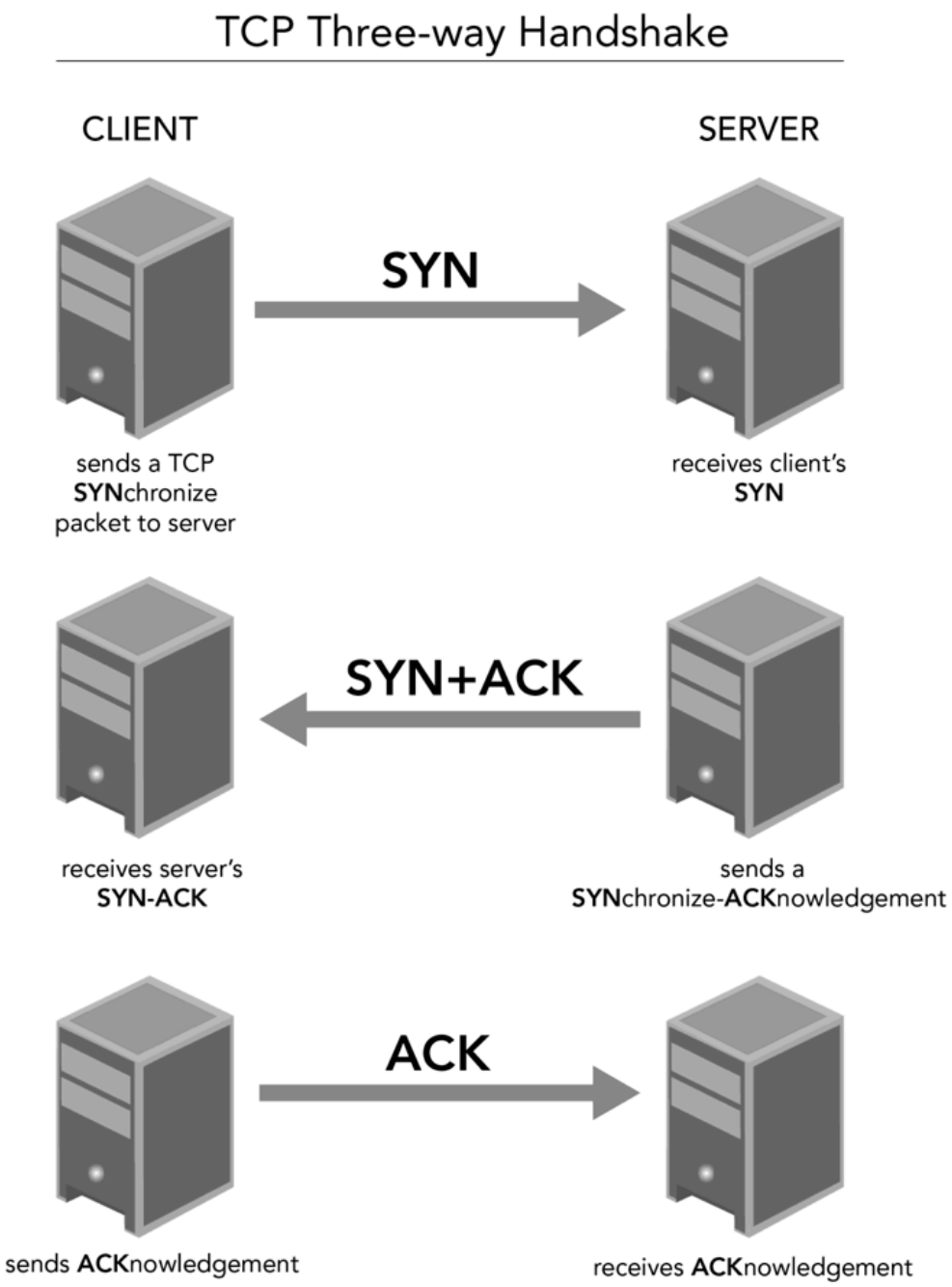
TCP Ports

TCP uses ports to match a communication with a service running on the computer. Each packet contains a source port and destination port. The source port is used to communicate with the service on the sending computer. The destination port is used to communicate with the service on the destination computer. The ports' range is 0-65535, with ports 0-1024 reserved for common network protocols such as HTTP (TCP port 80). TCP ports are different from UDP ports. For example, UDP port 80 points to a different service than TCP port 80.

Three-way Handshake Process

- The computer seeking the service assumes the role of the client
- The computer receiving the request assumes the role of the server

1. The client sends a connection request for a particular service (or port) to the server. The TCP packet has the synchronization (SYN) flag set. No data is sent throughout the TCP handshake.
2. The server responds by acknowledging (ACK) its receipt of the client's request and sending a synchronization (SYN) request of its own to ensure that the client is able to receive its transmissions. This SYN/ACK packet is sent back to the client.
3. The client acknowledges its receipt of the server's SYN/ACK by sending an ACK segment back to the server. The connection is now established.
4. Data can now be sent between the client and server.
5. Either the client or server can end the conversation by sending an RST packet or a FIN packet, or by ceasing communication.



How Does This Affect Network Monitoring?

Take a sample of network traffic when first setting up a monitoring device to ensure communications to and from the suspect system(s) are being captured. To confirm correct placement of the monitoring device and capture of suspect traffic, understanding and recognizing how a connection is initiated and established is important.

To determine quickly if the appropriate suspect traffic is being captured, look at three-way handshakes between the suspect system(s) and others. Also, determining when a communication session begins is crucial to isolating the periods of possible compromise. The three-way handshake will provide that information. Monitor placement assessment and analysis of sample traffic are discussed in-depth in the next module.

UDP

Along with TCP, UDP is a major transmission protocol used to transmit information. UDP performs the same functions as TCP but without the session control and related options. UDP does not maintain any sense of communication state. For the state to be maintained in a session occurring over UDP, it must be tracked by the Application layer protocol. Unlike TCP, UDP does not use any flags.

UDP is the protocol used for data streaming, such as streaming audio and video content. If packets are not received, the stream is not interrupted to retransmit the missing packets.

Error Control

UDP has some simple error control that is managed by the OS. If a packet is received on a closed UDP port, most OSs will send out an Internet Control Message Protocol (ICMP) Port Unreachable packet. However, since most firewalls will drop all ICMP packets, this error message may not reach its destination.

Web Protocols

Web protocols are the rules of communication for web browsing and related activities.

HTTP

HTTP is the most common protocol for web browsing traffic. This protocol governs the format for URL requests made by web clients and for fulfillment of those requests by a server. Items of value that can be obtained from transmissions using this protocol include URLs, page content and user actions such as posting of messages or other data.

Port: TCP 80

HTTPS

Secure HTTP (HTTPS) is the version of HTTP that uses encryption to keep outsiders from viewing the content of a web browsing session. This is done by encrypting the session with another protocol named Secure Sockets Layer (SSL). Because the traffic is encrypted, only information of minimal value can be obtained using this protocol.

Port: TCP 443

Markup Languages: HTML

Web pages exist in a format called markup language. This means that the pages contain not only content such as text and images but also instructions on how to format that content so that it appears a certain way on the page. These instructions, called tags, are transmitted in clear text along with the page content. The latest version of Hyper Text Markup Language (HTML) is HTML5.2, which includes improved support for geolocation and embedded objects such as video, music and bitmaps.

Email Protocols

Email protocols are rules for communication that govern electronic mail transmission. Items of value that can be obtained from transmissions using these protocols include email addresses, email server host names, transmission times and message content.

SMTP

SMTP is responsible for the transfer of email messages from the sender to the server and between servers as it is moved along the net to the recipient's mail server. SMTP handles the complex details of email transfer without intervention by the user.

A person will typically use a program to create a message. Programs used are either a client program, such as Microsoft Outlook, or a web browser interface. The user's program sends the outgoing message to the outgoing mail server, or SMTP server. The message is then forwarded to the recipient's mail server (possibly with some intermediate handoffs), where it will reside until the recipient's program uses either POP3 or Internet Message Access Protocol 4 (IMAP4) to retrieve it.

Port: TCP 25

POP3

Post Office Protocol 3 (POP3) is used to retrieve email from an email server. It is defined by Internet Engineering Task Force (IETF) document Request for Comments (RFC) 1939. POP3 was designed for use with an intermittent connection. A user's email client, when connected to the network, would use POP3 on the local machine to check the server for new messages. The messages would be downloaded to the local system, allowing messages to be read even when the local machine is not connected to the network. Because of this design, POP3 does not permit manipulation of the messages on the server other than downloading and deleting.

At its inception, POP3 was defined to hold messages until they are collected by the user's email client. Today, most email programs can be set to either retrieve messages and delete them from the server or retrieve messages and leave them on the server.

Port: TCP 110

IMAP4

IMAP4 is used to access email on an email server and is defined by IETF document RFC 3501. The IMAP4 rules differ from POP3. With IMAP4, the user can manipulate the messages on the server. Rules are defined allowing the email program to create, delete and rename mailboxes, check for new messages, permanently remove messages, and search messages.

Another major difference with POP3 clients is that they connect to the server briefly just to download messages. Using IMAP4, the connection between client and server remains active as long as the interface is active. Consequently, for users who have many messages or large messages, IMAP4 results in faster response times.

When email is accessed via a web browser interface, IMAP4 is typically used, but IMAP4 is also supported by the popular email clients as well. When using a client to access IMAP4 mail, two modes can be used: offline or online. web browser interfaces always operate in online mode.

When in online mode, the parts of a message are only downloaded to the local program as they are demanded. Therefore, when a user checks for new mail, only the mail headers are retrieved from the server. As the user chooses to read messages or view attachments, they are downloaded at that time.

Offline mode simply downloads the requested contents of the mailbox, including the headers, body and attachments, to the local client. This allows the user to access the mail when the machine is not connected to the network. Any changes the user makes to the mail on the local system in offline mode will not be reflected on the server.

With IMAP4, the server is considered the authority. All messages on the local system are treated like temporary cache files. Thus, when messages are deleted from the local mail client in offline mode, they will still exist on the server.

Port: TCP 143

Web Mail

Many people also use web-based email systems. For these, web protocols such as HTTP and HTTPS are used instead of the typical mail protocols listed previously.

Chat Protocols

Chat protocols are used for real-time, text-based communication between individuals on separate computers. Valuable information that can be obtained from transmissions using these protocols includes chat handles (names), IP addresses and the content of any discussions.

Internet Relay Chat

Internet Relay Chat (IRC) is an older chat program that is still commonly used. The IRC revolves around chat rooms where multiple individuals can communicate. IRC also offers direct messaging between individuals and file transfers.

Port: TCP 6665-6669

File-sharing Protocols

File-sharing protocols are used for the transfer of files from computer to computer. IP addresses, host names, filenames, and file content are types of information obtained from these protocols.

File Transfer Protocol

File Transfer Protocol (FTP) is a set of rules for transferring files that is commonly used on the internet. FTP commands are transferred over TCP port 21, but data transfers typically occur over TCP port 20.

Ports: TCP 21 and 20

NetBIOS

Network Basic Input/Output System (NetBIOS) is a protocol that performs network session construction and maintenance for Microsoft Windows computers. Typically, a NetBIOS session is initiated between the computers sharing the file before Windows file sharing takes place. Several subcomponents of NetBIOS exist, and they use different ports:

- **NetBIOS Name Service:** TCP port 137
- **NetBIOS Datagram Service:** UDP port 138
- **NetBIOS Session Service:** TCP port 139

SMB/CIFS

Server Message Block/Common Internet File System (SMB/CIFS) is Microsoft's file-sharing protocol. It is the standard protocol used to share out files and folders in the Microsoft Windows OS. This protocol was created and is maintained by Microsoft, but it can be used by some UNIX applications.

Port: TCP 445

Note: Some implementations of SMB still transfer files over TCP port 139.

Network File System

Network File System (NFS) is a file-sharing protocol used primarily by Sun Solaris as well as other versions of UNIX and Linux.

Port: TCP 2049

Remote Procedure Call

Remote Procedure Call (RPC) programs are used to provide a network presence for applications that are either not network-enabled or that need another program to act as the first point of contact for communication. For example, to communicate with NFS on a Solaris server, a file-sharing client must first contact RPC to determine the appropriate method of communicating with the file-sharing application.

Port: UDP 135 (Windows)
Port: UDP 111 (Solaris)

Name Resolution Protocols

Name resolution protocols provide a mechanism for resolving a machine name to a machine address. This is useful because it is easier for people to remember names than the numbers that make up a network address. Useful information that can be obtained from these protocols includes the name-to-address mappings of devices on a network.

DNS

Domain Name System (DNS) is the standard for name resolution on the internet. DNS provides a mechanism for the translation of domain names such as Microsoft.com to IP addresses, and the reverse.

Port: UDP 53

Domain Names and DNS

Domain names provide a system that is easy for humans to read and helps locate certain resources available on the internet. The system correlates these resources with the organization, group or individual that is making them available. This allows resources to be made available without regard to their actual physical location and without relying on difficult-to-remember IP addresses. The use of domain names and DNS also allows for the reallocation of IP addresses to different resources by their registered owners without breaking the links that point to the internet-based resources.

To make internet destinations easy to remember, most networks and websites use text-based domain names, such as www.dc3.mil, rather than IP addresses. Because the internet is based on numerical IP addresses, the internet DNS translates textual domain names into numerical IP addresses to create an internet connection. For example, when a user types the web address to a favorite site, DNS receives the resource request and translates it into the correct corresponding IP address.

Top-level Domains

The group of letters to the right of the far right period (.) in a domain designation is called the top-level domain (TLD). The TLD was originally intended to describe the general type of organization to which the domain was assigned. Some TLDs have certain conditions that must be met for their use.

Several types of TLDs exist. The most commonly encountered TLDs are for general use where registration is available to anyone. These TLDs often display a two-character country code indicating a specific country or territory. Keep in mind, however, that a website with a specific country code does not always have to reside in that particular country.

The following describes the typical application of general-use TLDs:

Commercial Use

If using the internet for business or profit or reselling internet service, an entity will most likely be assigned a .com or .net grouping. For example, IBM has distinct presences on the internet. The corporation and employees use ibm.com to conduct business. Customers also use this address to download instructions or a driver for an improved piece of computer hardware.

In addition, Comcast or EarthLink has an internet Service provision business, known as comcast.net or Earthlink.net, where customers buy internet, email and newsgroup access.

Educational Use

Educational institutions are traditionally assigned the .edu grouping. Examples of educational domains include mit.edu and lauds.k12.edu.

Governmental Use

Military and civilian government agencies use and give their employees access to the internet for research and communication. These domains usually contain .mil for military and .gov for civil government. Examples include fbi.gov or navy.mil.

Nonprofit Organization Use

Nonprofit organizations typically use .org in the domain. Examples include pbs.org or missingkids.org. Public organizations, such as libraries, also use .org.

Second-level Domains

Within the DNS hierarchy, a second-level domain is the designation immediately to the left of the far right period (.). This name is generally representative of the organization that registered the domain with a domain name registrar. In some cases, particularly when country-specific TLDs are used, the SLD name may represent the organization type and not the specific organization itself. In those instances, a third-level domain name is used to represent the organization.

Subdomains

Subdomains, also known as child domains, are segmented portions of a larger domain. A single subdomain may cover only a specific portion of the content of a domain. For example, the domain dc3.mil may have subdomains for DCCI, DCFL and DCITA. The full domain names, respectively, would be dcci.dc3.mil, dcfl.dc3.mil and dcita.dc3.mil.

Accessing a subdomain may direct clients to different locations on a server or to different servers.

Although they appear to be the same, it is important to distinguish between subdomain names and host names on the internet. Host names are domain names assigned to specific host devices. Subdomains identify a portion of the larger domain and may contain a large number of actual hosts.

How DNS Works

The internet DNS is best represented as an inverted tree structure. The tree is subdivided into distinct zones beginning at the root zone, with each zone being served by an authoritative name server.

At the top level of DNS is a single root zone, administered by a set of 13 root name server clusters and distributed in various geographically diverse locations around the world. The root zone includes information related to all top-level international, country-code, and generic top-level domains. A root name server answers requests directed to the DNS root zone and redirects requests for a particular TLD to that TLD's assigned name servers. When a DNS server is queried, it can respond in one of the following ways:

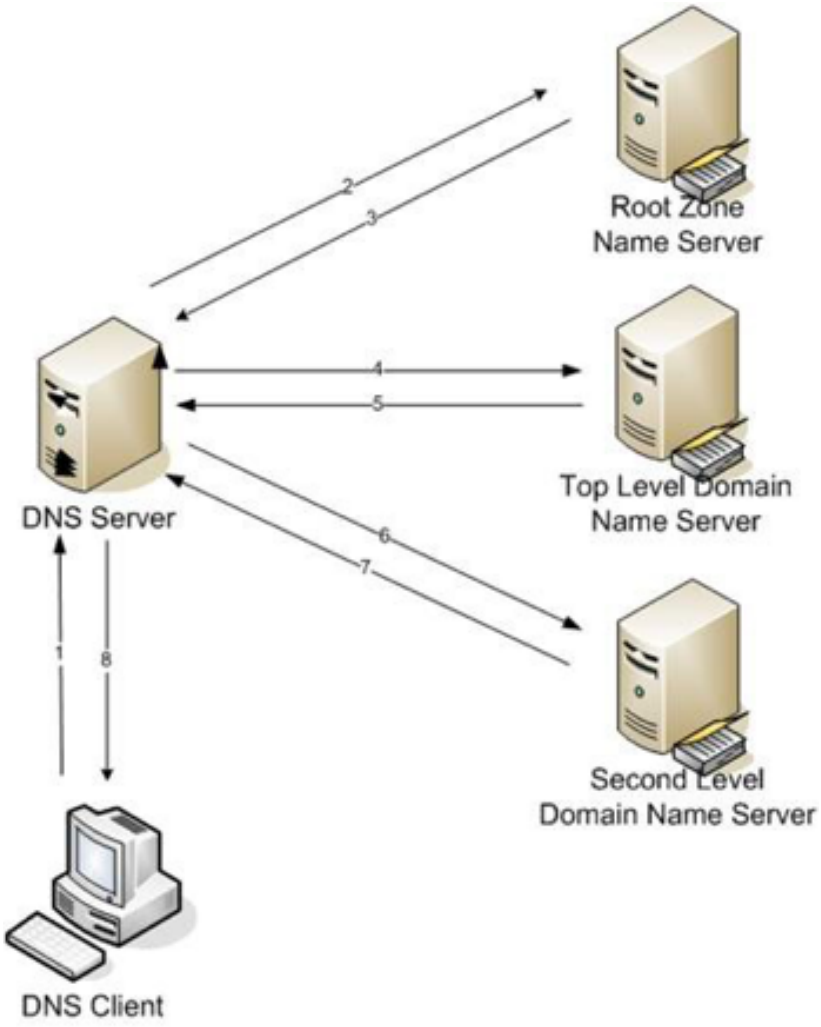
- Respond to the request directly by providing the requested information
- Provide a pointer (referral) to another DNS server that can assist in resolving the query
- Respond that the information is unavailable
- Respond that the information does not exist

The request is submitted and forwarded through the DNS until the requested domain name is resolved to the correct IP address or a response is received that the information is unavailable or does not exist.

These steps describe the typical route taken by DNS to resolve a domain name, such as www.dc3.mil.

- The DNS client submits a DNS query to its designated DNS server, i.e., www.dc3.mil.
- Assuming that the address requested is not in its existing cache, the DNS server refers the request to a root zone server.
- The root zone server provides the requesting DNS server the IP address of the appropriate TLD name server, i.e., .mil.
- The DNS server forwards the request to the TLD server for resolution.
- The TLD server responds back to the DNS server with the IP address of the authoritative name server for the requested domain, i.e., dc3.
- The DNS server then forwards the request to the SLD name server for resolution.
- The SLD name server responds back with the IP address of the specific subdomain (or host name) being requested, i.e., www.
- The DNS server forwards the IP address of the fully resolved domain name, i.e., www.dc3.mil, back to the DNS client.

The typical route taken by DNS to resolve a domain name, such as www.dc3.mil.



DNS Lookups

The IP address of a specific domain name can be directly submitted to the DNS. A DNS lookup involves sending a DNS query for a specific domain name and retrieving the IP address of the identified server. A user can perform a DNS lookup by using the nslookup command in Windows and Linux.

The following text shows an example of a DNS lookup:

```
C:\>nslookup www.dc3.mil
Server: nsbalt.bellatlantic.net
Address: 151.196.0.38

Name:  dc3.mil
Address:  214.3.152.67 Aliases:  www.dc3.mil
```

A reverse DNS lookup involves sending an IP address to a DNS server and retrieving the domain name of the server. The following text shows an example of a reverse DNS lookup:

```
C:\>nslookup 214.3.152.67
Server: nsbalt.bellatlantic.net
Address: 151.196.0.38

Name:  NS1.DC3.MIL
Address:  214.3.152.67
```

Note: DNS queries can also be performed with web-based utilities that will provide the same information. Some options include:

- network-tools.com
- CentralOps
- www.dnsqueries.com/en
- www.robtex.com

Using these utilities hides the origin of the request from the DNS administrators.

Procedure: Finding a Website's IP Address With CentralOps.net

Network tools as well as several websites, such as CentralOps and Robtex, can be used to look up a website's IP address. Some of these websites contain restrictions for the number of queries that can be placed within a specified amount of time. Follow these steps to find a website's IP address with CentralOps.net.

Step	Action
1	Open Internet Explorer and go to the following URL address: centralops.net
2	In the main page, under the list of "Free online network utilities," click the AutoWhois link. In the AutoWhois window, in place of excite.com, type: csc.com
3	Click the Go button.
4	Look at the answer provided by CentralOps.net for the WHOIS query.
5	Close Internet Explorer.

Note: To look up .mil sites, checking www.nic.mil from a .mil site is necessary. This restriction keeps that domain more protected than it once was.

Procedure: Determining a Website's IP Address With a Firefox Plug-in

Several plug-ins for Firefox allow a user to view in real time the IP address of the website the user is visiting. When using these plug-ins with Firefox, keep in mind that you are directly accessing the website. The session will likely be recorded on the web server you are trying to identify.

Remote Command Line Protocols

Remote command-line protocols are used for network sessions where a user on one computer has command-line control of another device. These protocols are typically used for remote administration. Information that can be obtained from remote command-line sessions includes usernames and passwords, the commands typed by the user, and the responses received.

Telnet

Telnet is a common remote command-line program and protocol. All commands and responses transmitted via Telnet are sent in the clear. Commands are typically sent one letter per packet.

Port: TCP 23

SSH

Secure Shell (SSH) is an encrypted remote command line protocol that can also be used for sending encrypted files. Little useful information can be obtained from an SSH session because the content is encrypted.

Port: TCP 22

Lesson 4

Packet Headers

A packet is the basic unit that carries data over the network. Packets contain headers, which contain information about the packet. This information is required for the packet to reach its destination and includes source and destination, timestamps, and other data.

This lesson introduces packet headers and how to analyze them.

OBJECTIVES

After completing this lesson, students will be able to:

- Analyze packet headers

IP Header

The IPv4 header is concerned with determining the path in which a packet will travel from one network to another.

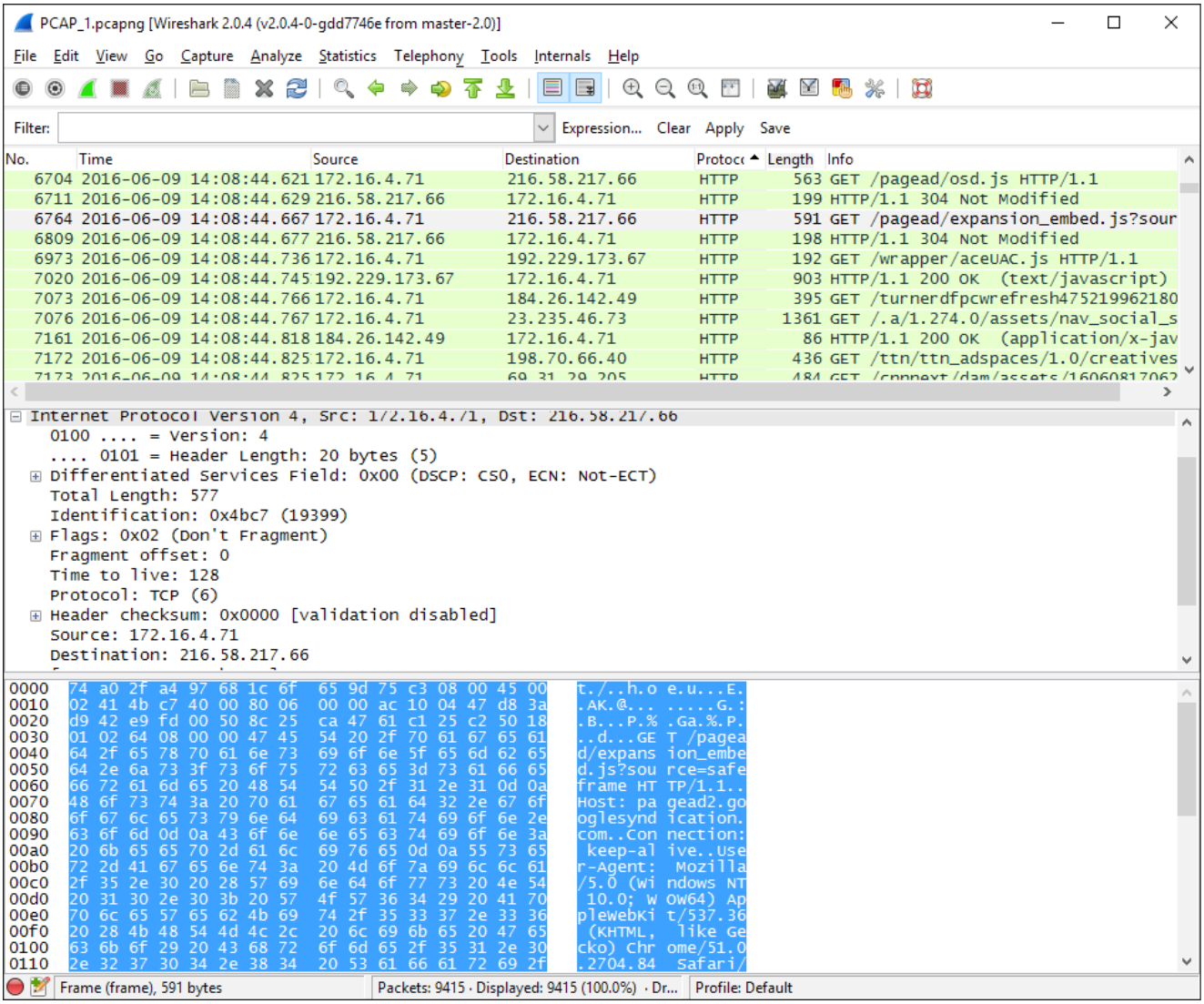
IP HEADER				
0 32				
Version	Header Length	Differentiated Services Field	Total Length	
Identification			Flag (3-bits)	Fragment Offset
Time to Live		Protocol	Header Checksum	
Source IP Address				
Destination IP Address				
Options & Padding (if any)				
Data				

Field	Description
Version (4 bits)	Version of the IP protocol.
Internet Header Length (4 bits)	Length of the header, typically 20 bytes.
Differentiated Services Field (formally, the Type of Service [TOS] octet) (8 bits)	Services intended to provide a framework and building blocks to enable deployment of scalable service discrimination in the internet. **Differentiated Services Codepoint (DSCP):** A specific value that should map to specific, standardized per-hop behavior.* **Explicit Congestion Notification (ECN):** Indicates the existence of traffic congestion. ECT – ECN-capable Transport CE – Congestion Experienced * Per-hop behavior is a description of the externally observable forwarding treatment applied at a differentiated service-compliant node to a behavior collective.
Total Length (16 bits)	Total length of the IP datagram, including the transport header (TCP/UDP) and data.
Identification (16 bits)	An integer value used to identify all fragments of a datagram; should be unique for each new datagram sent by a host.
Flag (3 bits)	Determines if the packets will be fragmented as they are transmitted from one network to another. **Reserved (R)** **Don't Fragment (DF)** 0 = Fragment if necessary 1 = Do not fragment **More Fragment (MF)** 0 = This is the last fragment 1 = More fragments to follow
Fragment Offset (13 bits)	Used with fragmented datagrams to indicate the position the data in this fragment occupies in the original message.
Time to Live (8 bits)	TTL is set by the datagram sender, is used to track the lifetime of the datagram, and is decremented by routers as the datagram passes through them.
Protocol (8 bits)	The Transport layer protocol carried by this datagram 1: ICMP 6: TCP 17: UDP 41: IPv6 over IPv4 89: OSPF
Header Checksum (16 bits)	An algorithmic calculation used to ensure the integrity of the fields in the IP datagram
Source IP Address (32 bits)	The source's IP address
Destination IP Address (32 bits)	The destination's IP address
Options	Supports debugging, measurement and security facilities
Padding	Used if Options are transmitted to ensure a 32-bit word

Wireshark is an open source GUI network packet analyzer that captures network traffic data packets and displays them for the user to browse interactively or from a previously saved capture file.

Procedure: Follow these steps to view the network traffic and analyze packet header information via Wireshark

Step	Action
1	Double click on the Wireshark icon on the desktop
2	Click Open under the Files menu option in the main window. Select a file as directed by the instructor. The file may take some time to load.
3	Click Protocol to sort the list alphabetically by protocol type.
4	Select a frame
4	Click the plus symbol to view the Internet Protocol header information
5	Click the plus symbol to view the Transmission Control Protocol header information



ICMP Header

ICMP is an extension to the Internet Protocol. It is used to report errors in IP datagram routing. ICMP uses a fixed message format that has pre-defined message codes to convey information.

ICMP Message		
0 32		
Type	Code	Checksum
Unused		
IP Header + 64 bits of original data		

Field	Description
Type (8 bits)	Defines the category of ICMP message. The next table provides better detail.
Code (8 bits)	When applicable, provides more specific information to the Type category.
Checksum (16 bits)	A basic check on the message.

ICMP Message Types/Codes

The Request for Comments (RFC) assigns specific numbers to ICMP messages types and codes. The following table provides a listing:

Type	Name	Code
0	Echo Reply	
1	Unassigned	
2	Unassigned	
3	Destination Unreachable	0 Net Unreachable 1 Host Unreachable 2 Protocol Unreachable 3 Port Unreachable 4 Fragmentation Needed and Don't Fragment was Set 5 Source Route Failed 6 Destination Network Unknown 7 Destination Host Unknown 8 Source Host Isolated 9 Communication with Destination Network is Administratively Prohibited 10 Communication with Destination Host is Administratively Prohibited 11 Destination NetworkUnreachable for Type of Service 12 Destination HostUnreachable for Type of Service
4	Source Quench	
5	Redirect	0 Redirect Datagram for the Network (or subnet) 1 Redirect Datagram for the Host 2 Redirect Datagram for the Type of Service and Network 3 Redirect Datagram for the Type of Service and Host
6	Alternate Host Address	
7	Unassigned	
8	Echo (used by ping)	
9	Router Advertisement	
10	Router Solicitation	
11	Time Exceeded	0 Time to Live exceeded in Transit 1 Fragment Reassembly Time Exceeded

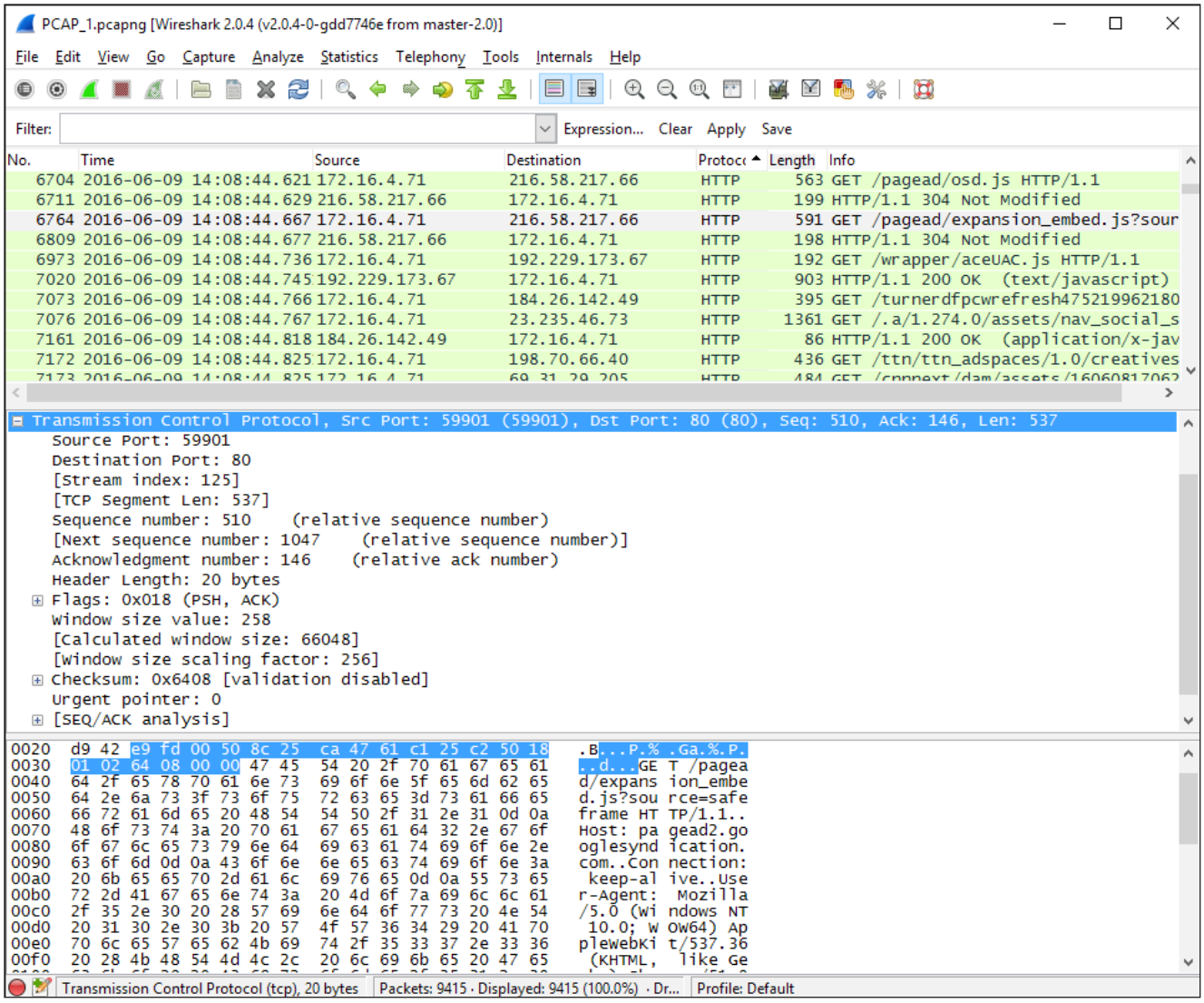
12	Parameter Problem	0 Pointer indicates the error 1 Missing a Required Option 2 Bad Length
13	Timestamp	
14	Timestamp Reply	
15	Information Request	
16	Information Reply	
17	Address Mask Request	
18	Address Mask Reply	
19-29	Reserved	
30	Traceroute	
31	Datagram Conversion Error	
32	Mobile Host Redirect	
33	IPv6"Where-Are-You"	
34	IPv6 "I-Am-Here"	
35	Mobile Registration Request	
36	Mobile Registration Reply	

TCP Header

For services that require high-reliability, the Transport layer uses a connection-oriented protocol, TCP.

TCP HEADER										
0 32										
Source Port Address										Destination Port Address
Sequence Number										
Acknowledge Number										
Header Length	Reserved	CWR	ECN	URG	ACK	PSH	RST	SYN	FIN	Window Size
Checksum										Urgent Pointer
Options & Padding (if any)										
Data										

Field	Description
Source Port (16 bits)	The source port number
Destination Port (16 bits)	The destination port number
Sequence Number (32 bits)	Once connection initializes, both nodes agree to the new sequence numbers and the connection starts. Sequence numbers count segments as they are transmitted and received.
Acknowledgment Number (32 bits)	Response numbers to correct receipt of sequence numbers (packets).
Header Length (Data Offset) (4 bits)	Measures the offset to the start of the application data field.
Reserved(4 bits)	Reserved field
Control Bits (Flags) (8 bits)	**CWR** The Congestion Window Reduced field informs the data receiver that the congestion window has been reduced. **ECN** The Explicit Congestion Notification indicates the existence of traffic congestion. **URG** Indicates that the urgent pointer field is valid. This field points to an octet in the data field that is the end of urgent data. ACK Indicates that the acknowledgment field is valid; acknowledging the receipt of a transmitted packet. **PSH** The push flag causes the remote TCP layer to pass this segment immediately to the application layer. **RST** The reset flag indicates that an error has occurred and the connection should be forcibly closed. **SYN** The synchronize flag is used at the beginning of connection setup between two nodes. **FIN** The FIN flag is used to end connections. **Note:** TCP controls the flow of network traffic by increasing or decreasing the window size. Yet when TCP traffic is flowing freely, occasional network congestion can occur, resulting in routers dropping packets. However, network nodes that are ECN-capable may receive a Congestion Experienced (CE) bit from a router, advising the source node to reduce its congestion window.
Window Size (16 bits)	The sender advertises the amount of buffer space this node has allocated to this connection and the amount of data the sender is willing to accept.
Checksum (16 bits)	An algorithmic calculation check on the header and data.
Urgent Pointer (16 bits)	The value in this field points to the end of data field that is considered urgent and requires immediate attention.
Options	**NOP** (No Options) **SACK** (Selective Acknowledgments) **Maximum Segment Size** (MSS) **MSS:** 536= Serial 1460 = Ethernet 4460 = FDDI (fiber)
Padding	Used when options are transmitted to ensure 32 bits.



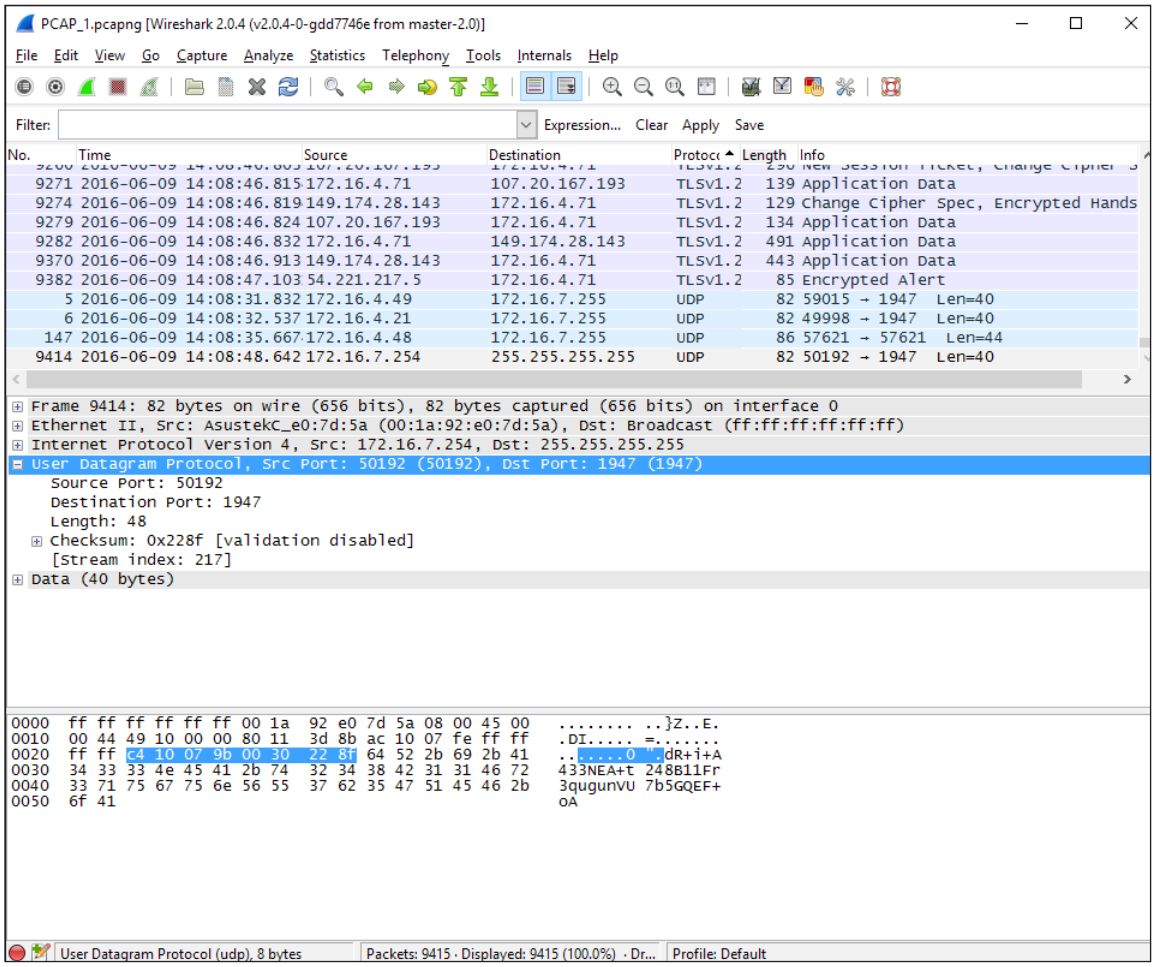
A Wireshark example of a HTTP packet selected. In this example the traffic uses TCP in the transport layer with source port 59901 and destination port 80.

UDP Header

UDP provides a best-effort (connectionless) service for the transfer of individually addressed datagrams.

UDP HEADER	
0 32	
Source Port	Destination Port
Length	Checksum
Data	

Field	Description
Source Port (16 bits)	The source port number
Destination Port (16 bits)	The destination port number
Length (16 bits)	The number of bytes in the UDP packet (including header)
Checksum (16 bits)	A basic check on the header and data



A Wireshark example of a UDP packet selected. A UDP header packet is small only 8 bytes and only has four parts.

MODULE 2

Networks and Witness Devices

A basic understanding of network devices is necessary to effectively monitor a network. This module details the common forms of network traffic and how to identify and interpret network log content.

OBJECTIVES *After completing this module, students will be able to:*
Describe the data that can be recovered from common witness devices
Perform logical and physical assessments of a network to identify potential witness devices
Define network logs
Describe common network log content

Lesson 1

Understanding and Gathering Network Logs

A network log is any file that contains information relating to events that have occurred on a computer network. Investigators must be familiar with different types of network logs to perform a proper investigation. This lesson introduces common forms of network logs and explains how to identify and interpret network log content.

OBJECTIVES

After completing this lesson, students will be able to:

- Define network logs
- Describe common network log content

What Are Network Logs?

Network logs are transactional records generated by network devices for the purpose of tracking information and diagnostics.

Network logs are generated from multiple devices, ranging from applications and systems to network communication devices. These devices can log information about the network communications in which they take part.

Network-based logs leave an audit trail that can be used to reassemble or reconstruct network activity. Logs generated from network devices can provide critical information as investigators try to determine exactly what occurred on the network or host.

Network logs can be generated from:

- Routers
- Firewalls
- Switches
- Proxies
- Intrusion detection systems (IDS)
- Sniffers
- Servers

Text-based vs. Binary Logs

Logs are typically found either in text or binary format. This distinction is important because the format of the log will dictate how analysis is conducted.

Binary Logs

Binary logs are any logs that include unformatted or raw data. They may include some text, or they may be encoded entirely in a non-text format. Binary logs are generally of two types:

- A log recorded in a proprietary binary format by a vendor so that the log can only be viewed using the vendor's software
- A binary record of traffic that traversed a network connection

In either case, binary logs cannot be read using a standard application. Proprietary binary logs must be read by an application that understands the proprietary format. Network captures must be read by a program that can perform at least rudimentary protocol analysis, such as reading network protocols to extract and display protocol header elements and encapsulated user data.

Text Logs

Many logs are recorded in text format, and most text logs can be opened and read with a standard text editor. This may not always be feasible due to log size, but it is possible. Text logs generally only store various text-based information determined by the application that created them. Other non-textual data, like raw binary, is lost.

Common Network Log Content

Network logs record objective and subjective content. Objective information is raw data recorded from a network event. For example, a sniffer can record all network data received on a transmission medium. The sniffer does not interpret the data; it simply creates a record. This data is considered objective.

An example of subjective log data is an IDS log entry identifying a particular attack. In this case, the system interprets certain network data as a specific kind of attack. But this is not always the case since false positives can occur.

Objective log data can be used as direct evidence of a particular event. Subjective data must be verified through further investigation.

Network Log File Attributes

Network logs have different data included in their content. This data provides the information needed to reassemble and analyze incidents. Some common network log file data content includes:

- **Timestamp:** The time the message was sent or received
- **IP address:** The IP address of the sender or receiver
- **Direction:** If the message was sent or received
- **Protocol:** Which network protocol was used to communicate the message
- **Port number:** The number assigned to either the sending or receiving application or transmission over a TCP/IP network
- **Message length:** The complete message length in bytes
- **Code or message:** The error, alert or informational code or message

Common Log File Extensions

Logs can be saved with several file extensions, including: .

dat	The DAT file type is primarily associated with 'Data'. Can be just about anything: text, graphic, or general binary data. There is no specific structure for a .DAT file. You can use an editor like EditPad Pro to look inside a .DAT file and possible determine its contents and relationship with a program.
.enc	Generic encoded file that may use one of several different types of encodings; helps protect the file from unauthorized use, or helps set up the file for a particular use, such as Internet transfer.
.log	A LOG file is a journal used by various operating systems and programs. It typically contains a plain text log of certain events with their timestamps. The file may be created by the operating system to keep track of system events or by a software installation program to list location and names of installed files.
.trn	Backup file created by Microsoft SQL Server, TRN files are used for database restores and may be used with other TRN files to chain roll back to any previous database state.
.apc	APC file is an A-Pac Compressed Audio. The Audio Packer (A-Pac) is an old lossless audio compressor.

.dgc	Archive created by DGCA (DigitalGCodecArchiver), a file compression utility created by Japanese developer Shin-ichi Tsuruta; saves one or more files in a compressed format and can compress at better ratios than the .ZIP format. Intuit TurboTax uses this file type.
.eth	ETH file is an Ethnograph Document. Ethnograph is qualitative data analysis software that allows you to create ethnographic projects.
.pkt	Contains a simulation of a network setup for Packet Tracer, a network monitoring program developed by Cisco Systems; allows users to test various network configurations; used primarily for educational purposes.
.txt	A TXT file is a standard text document that contains unformatted text. The file is recognized by any text editing or word processing program and can also be processed by most other software programs
.cap	Contains packets collected by a packet sniffing program; saves raw data captured over a data transmission; also called a trace file or bone file and is used by multiple packet sniffer applications such as Wireshark.
.dmp	A DMP file is a file that contains data "dumped" from a program's memory space. It is often created when a program has an error or crashes and may also be saved by the program "Savedump.exe" on the first reboot after a crash. The file is usually named "Memory.dmp."
.fdc	file used by AutoCAD, a program that allows engineers and designers to create 3D drawings; contains definitions for application fields; used for displaying information within the AutoCAD user interface.
.snoop	SNOOP file is a Snoop Verbose Trace File. Snoop is a flexible command line packet analyzer included as part of Sun Microsystems' Solaris operating system.

Lesson 2

Witness Devices

Many computers and devices can log information about network communications activities. Witness devices are different because they can record transmissions for which they are not the intended source or destination. This lesson introduces witness devices and their importance to a network investigation.

OBJECTIVES *After completing this lesson, students will be able to:*
Define the term witness device
List common witness device types, including switches and routers
Explain the differences between hardware- and software-based devices
Explain the ways to obtain data from witness devices

About Witness Devices

A witness device is any network device that has recorded data about network transmissions to or from the subject of the incident.

The analysis of an imaged device may not be sufficient to reconstruct the sequence of events that lead to a network incident. Information from witness devices may help you determine the routes and methodologies used by the attacker, as well as a timeline of the activity.

Logs from witness devices are often essential in determining the exact nature of an attack and for tracing the attack to its origins.

The witness devices discussed in this module are:

- Switches
- Firewalls
- Routers
- Sniffers
- Intrusion detection systems (IDS)
- Remote log storage servers
- Wireless access points

Hardware-based Devices

A hardware-based device, such as a firewall or router, is a computer built entirely to perform a certain function. The platform is specialized, from the case and components to the OS. These devices come in a variety of shapes and sizes and are almost always smaller than the average PC tower case.

The variety in appearance of hardware-based devices makes them difficult to identify. Fortunately, most vendors label their appliances to indicate their functions. For example, many firewall appliances have the word firewall printed on the case. For unlabeled devices, you can find the make and model on the exterior of the case and research it to identify the type.

Hardware-based devices often keep most data in volatile storage. Because of this, obtaining data from these devices without turning them off is important. When the devices are powered down, much of their valuable data will be lost.

Software-based Devices

A software-based device is a program that performs a specific function and is loaded on normal PC hardware with a standard OS. For example, a desktop PC with Windows can also have a firewall program loaded on it. The computer acts as a firewall, even though it wasn't originally built for that purpose.

Because they could resemble an average PC, software-based witness devices are difficult to identify. Some system administrators label their computers by function, but often do not. You may have to reference the computer against documentation for the network to discover its function. The placement on the network also can indicate the device's purpose.

Methods of Gathering Data

Four general methods are used for gathering data from witness devices. Although implementing the device for the various methods may differ, any can be used to extract data.

Method 1: Direct Console Connection

A direct connection can be made to some devices via a cable from a workstation. The connection is usually either USB- or serial-based and provides administrative access to the device.

Method 2: Remote Terminal Connection

Although most switches are Network Access devices, they can provide internet routing as well. Switches that combine network access switching and Internet layer routing are called multi-layer switches.

- Telnet
- Secure Shell (SSH)
- rlogin

Method 3: Web Browsing

Some devices allow web-based, remote administration for ease of use. If this is the only option, use a web browser to connect to the web-based interface.

Method 4: Standard Media Imaging

For software-based devices, imaging methods are still an option. If the device can be powered down, the hard drives can be duplicated using your organization's standard imaging techniques.

General Considerations

Switches can also provide the switch port analyzer (SPAN) port capabilities that are needed to capture network traffic to a sniffing device. Recall from the network access layer of the TCP/IP model that switches learn the MAC address of the devices attached to them and route traffic among the devices that want to communicate. No other devices can intercept that conversation unless the switch has a SPAN port configured to capture other traffic.

- Hardware-based devices use mostly volatile data storage. If you turn them off, you may lose valuable information.
- Consider the position and importance of a device in the network before powering it down for imaging. For example, turning off a firewall that guards a network from the internet will disrupt access for every user who sits behind it. In some cases, you may need to obtain permission before powering down such a device.
- You usually must obtain an administrative username and password to access a device.
- Because logs are the records of activity maintained by witness devices, they are the primary target of data gathering. However, you need the device's full configuration to know how the device operates on a network and whether the device has been compromised. Obtain device configuration information.
- Witness devices do not always store their logs locally. Some may be configured to send logs across the network to a server. When gathering

data from witness devices, always check for remote log settings and determine whether you need to expand your data-gathering activities to other devices.

- Network environments can be large and complex. To accurately determine the scope of an incident and identify the location and function of related witness devices, obtain a copy of all network documentation. This can include:
  - Topology map of the network
  - List of devices and their functions
  - Device/server configuration settings
  - Network policies and procedures
- Creating hash values for the data collected is essential for establishing its validity for use later as evidence in court.

Device Configuration

The configuration information of network devices can show how the devices are configured on the network, which may aid an investigation.

This chart shows information that may be contained in a device configuration analysis:

Configuration information	**Importance**
Current configuration	This can be critical especially if detailed network diagrams are unobtainable. Configuration information can be used to glean information on traffic flows and how data is processed.
Past configuration	Past configuration can be analyzed to determine whether configuration issues are pertinent to an intrusion.
System user information	User configuration may reveal who is an authorized user of a device and the presence of authentication solutions such as RADIUS or TACACS.
Date/time settings	These can be used to correlate time during an investigation.
Access control lists (ACLs)	These configuration details can explain what communications are allowed on a network.
Syslog settings	Network devices can be configured to send all log entries to a syslog server.
SPAN ports	These are used to configure a network sniffer.
Database logging	Some network devices can send their log files to databases.

Lesson 3

Switches

A switch is one of many witness devices that can contain information relevant to an investigation. This can include the status of a port, the security of a port, the media access control (MAC) address, and logging information. For wired local area networks (LAN), the switch will often be the connection point for devices. Investigators should inquire about any switches when reviewing the network documentation.

This lesson introduces switches and their importance to a network investigation.

OBJECTIVES *After completing this lesson, students will be able to:*
Describe a switch
Explain the general purpose and function of a switch
Recognize a switch on a network diagram
Explain the types of data that might be obtained from a switch

About Switches

A switch operates like a bridge, but it includes more than two ports (up to 48 ports or more on large switches). Hosts on a switch communicate directly with one another without hosts on other ports receiving the communication. Like a bridge, a switch maintains a table of MAC addresses and their associated ports. Switches, like hubs, are the devices to which workstations, servers, IDSs, and firewalls connect directly. However, switches are faster, manageable, and can provide security functions.

Switches can provide evidence, but like routers, their evidence may not be as detailed as an IDS or a sniffer. Also, their information is usually volatile and can be lost if power is lost.

Switches can save their information locally or send it to another device such as a syslog server. A switch's configuration and any logging that can be

obtained can be used as evidence. Switches are often overlooked, but may be the only device that can provide information about a covert incident.

Configuration, Logging and Security

The configuration can provide information about the setup of the switch and its ports, particularly the port the suspect may have used. It may also have an ACL configured that can provide crucial information concerning whether a suspect may have had access to a device. If logging is activated, any violation of the ACL should be in the logs.

Although most of the configuration will be examined at the lab, knowledge of the configuration can help determine where to focus the investigation. For example, if you are presented with several switches and time is limited, a check of the configurations may help determine which switches you should focus on. This can save time and effort.

Spoofing MAC addresses is a common hacking behavior. A smart administrator will enable the logging functions on a switch to be able to capture MAC address and IP address associations. Logging will enable the administrator to identify the port to which each IP address was assigned.

If the suspect spoofed the victim's MAC address on his assigned port, the logs will indicate that action and the data will become evidence. On the other hand, if logging and port security are not configured, the switch will be of little use.

Switch Types

Some switches are link-layer devices, they can provide internet routing as well. Switches that combine link-layer switching and internet-layer routing are called multi-layer switches.

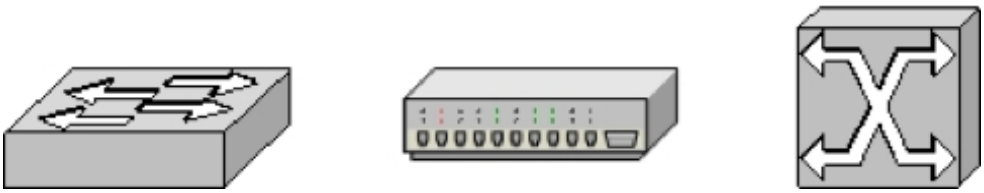
For example, the administrator presents you with the network documentation, which includes a diagram designating the placement of a router and a switch. These devices may provide a possible path for the incident. You bypass the switch because you think that it is just a link-layer device and you collect the data off the router instead. The examiner observes the router evidence and may make a determination that the suspect committed the offense. However, the defense, after looking at the evidence, performs a more thorough investigation of that same network and finds that the switch you did not collect from is actually a link-layer switch with Internet-layer routing, or a multi-layer switch.

The configuration collected shows an ACL rule being applied to the client's port on the switch that prevented access to the device in question. In short, network documentation may be misleading. You should either ask whether the switch is routing or get access to it and check whether it is routing.

Switch Port Analyzer Port

Switches can also provide the switch port analyzer (SPAN) port, also called mirrored port or monitored session, capabilities that are needed to capture network traffic to a sniffing device. Recall from the link layer of the TCP/IP model that switches learn the MAC address of the devices attached to them and route traffic among the devices that want to communicate. No other devices can intercept that conversation unless the switch has a SPAN port configured to capture other traffic.

On the other hand, hubs receive transmissions on one port and broadcast them out to all the others, which eliminates the need for a SPAN port. In short, a SPAN port is required for sniffers, IDS or other tools to monitor all traffic off of a switch.



Examples of icons commonly used to represent a switch on a network diagram. They may be labeled as switch, L1/L2 (Link layer or Internet layer) or they may be represented as these devices.

Log Examples

This prepares a CatOS-based switch to send syslog messages at facility local4. Also, the switch will only send messages with a severity of warning or higher. The syslog server is on a machine with an IP address of 192.168.0.30.

```
Console> (enable) set logging timestamp enable
System logging messages timestamp will be enabled.
Console> (enable) set logging server 192.168.0.30
192.168.0.30 added to System logging server table.
Console> (enable) set logging server facility local4
System logging server facility set to <local4>
Console> (enable) set logging server severity 4
System logging server severity set to <4>
Console> (enable) set logging server enable
System logging messages will be sent to the configured
syslog servers.
Console> (enable) show logging
Logging buffered size: 500
timestamp option: enabled
Logging history size: 1
Logging console: enabled
Logging server: enabled
{192.168.0.30}
server facility: LOCAL4
server severity: warnings(4
Current Logging Session: enabled
```

Lesson 4

Firewalls

Firewalls are devices used for perimeter network protection. If configured properly, firewall technology can provide extensive logs that may be valuable to an investigation.

This lesson introduces students to firewalls and their importance to a network investigation.

OBJECTIVES *After completing this lesson, students will be able to:*
Describe a firewall
Explain the general purpose and function of a firewall
Recognize a firewall on a network diagram
Explain the types of data that might be obtained from a firewall
Investigate and gather data from a firewall
Describe the function of a proxy server

About Firewalls

A firewall is a device used to guard the border between two networks. Firewalls are most frequently used to protect an internal network against access from the internet, which is generally not trusted.

Firewall rules allow traffic in or out of a network. Logging may be enabled to capture this traffic, which can be valuable during the analysis phase of a network incident.

How Firewalls Work

If a firewall is placed properly, all traffic entering or leaving a protected network must pass through it. Transmissions are checked using specified security criteria. The firewall blocks traffic that does not meet the criteria, allowing the rest to pass. This process is called filtering.

Firewalls have different filtering ability levels that work at different layers of the TCP/IP model. Firewalls operate at the Internet (IP addresses) layer, Transport (ports) and the Application (application data) layers. For example, a firewall that is Transport layer "aware" can filter traffic based on source or destination service ports, as well as source and destination IP addresses. Another firewall that works at the Application layer may filter requests for certain Web pages based on keywords in the URL.

Access Control Lists

The access control list (ACL) is the rule set that a firewall uses to filter each transmission. The administrator or engineer setting up the firewall will create a list of acceptable and unacceptable traffic based on criteria established by the organization. That list is the ACL, and the firewall will block any traffic the ACL is configured to deny.

A firewall is only as effective as its ACL configuration. Traffic the firewall is not configured to block will enter the network, no matter how damaging it may be.

Firewalls and Evidence

ACLs and logs can be obtained from a firewall.

- The ACL defines what the firewall is filtering. Some ACLs will even be configured to deny access to specific hosts on the internet that have previously scanned or attacked the network, providing a limited history of attacks.
- ACL violation logs contain information about every transmission that is denied because of a rule in the ACL.
- Traffic logs contain records of traffic that have successfully passed through the firewall, including source and destination IP addresses.

Firewalls may not be configured to log accepted traffic. In high-volume environments, storing and analyzing the volume of data may be too costly. So, if an attack gets through the firewall, the associated transmissions will not be recorded. Most firewalls only log failed communications attempts (ACL violations).

ACLs and ACL Violation Logs

ACLs and their corresponding violation logs may contain a variety of information depending on the abilities of the firewall. Some examples are:

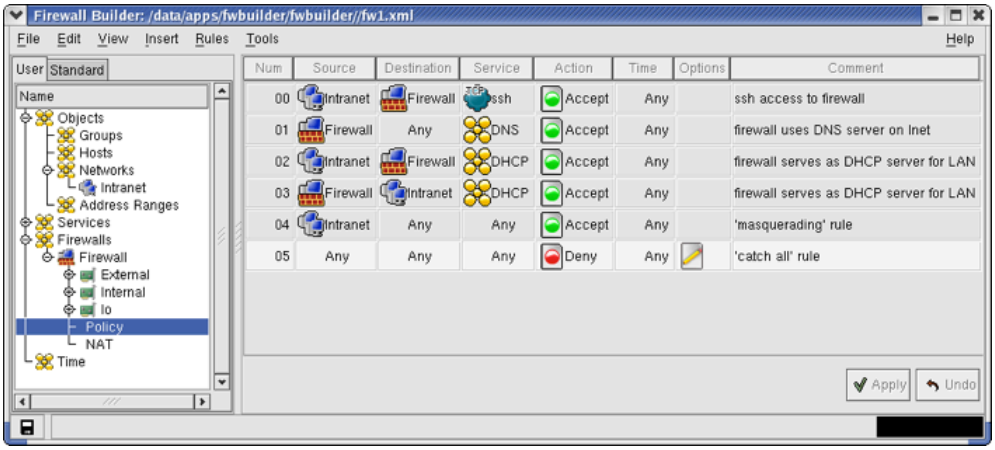
- Source IP, destination IP and port numbers
- Protocol-specific information, such as HTTP or e-mail headers, uniform resource locators (URL), usernames, and passwords
- The Application layer contents of a transmission. For example, some Application layer firewalls can scan the content of communications, such as e-mail or Web pages, for inappropriate words.
- Source and destination MAC addresses

Firewall ACL Example: Firewall Builder

An example of a firewall ACL follows. Each line in the ACL is called a rule. To reiterate, a rule specifies a set of criteria that includes source, destination, time, and type of traffic. It also specifies the action to be taken if a packet matches the criteria, which is to either accept or deny the packet.

The firewall checks a packet against each rule beginning with the one at the top of the list and proceeding to the bottom. When the firewall finds a rule that matches the packet, the action (Accept or Deny) specified by that rule is applied. This table describes the first rule of this ACL:

Data	Description
Intranet	Source of transmission; the internal trusted network
Firewall	Destination of transmission; the firewall itself
SSH	Type of traffic; SSH
Accept	Action to take; allow passage
Any	Accepted time for traffic; any
SSH access to firewall	A brief description of traffic; administration of firewall through SSH



A screen shot from Firewall Builder, an application used to construct ACLs for various types of firewalls.

Firewall Log Example: Cisco PIX

An example of a log entry from a Cisco PIX firewall appliance follows. It depicts an unauthorized transmission attempting to leave the network through the firewall. The packet was blocked because it matched a deny rule in the firewall's ACL. A detailed explanation of the entry is provided in the table.

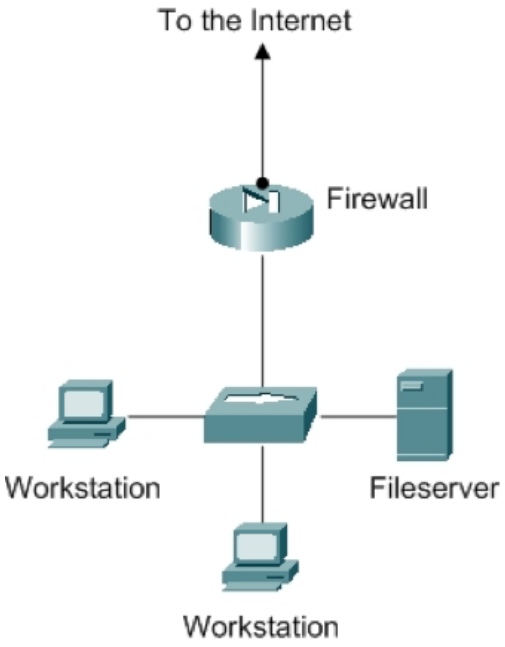
```
106023: Deny udp src inside:192.168.1.37/1028 dst
outside:192.168.2.20/161 by access-group
"inside_access_in"
```

Data	Description
106023	Cisco number identifying the event
Deny	Action taken by firewall
udp	Protocol of transmission
src inside	The transmission entered the firewall on the "inside" interface
192.168.1.37	Source IP address
1028	Source port number
dst outside	The transmission was directed at a network on the "outside" of the firewall
192.168.2.20	Destination IP address
161	Destination port number
by access-group "inside_access_in"	The name of the ACL that the firewall used in this action (Cisco uses the term access-group instead of ACL)

Locating and Identifying Firewalls

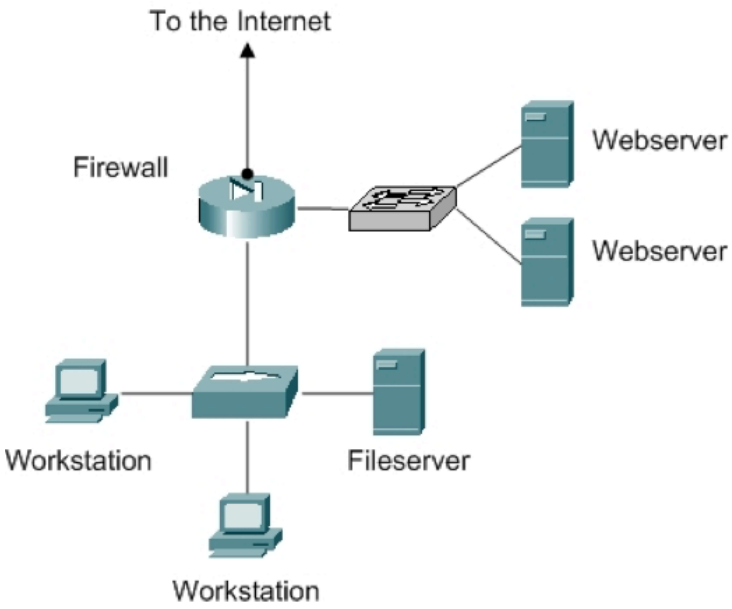
Firewalls are usually found at the border between two networks. The most common placement is between a LAN and the internet. However, firewalls can also be placed between two LANs and at other points on a network.

A simple network with two workstations and a server connected by a hub and protected from the internet by a firewall.



Network Placement: Demilitarized Zone Separation

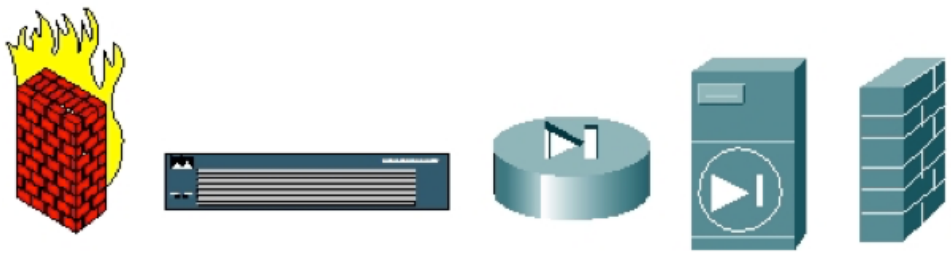
Some organizations require public access to their network. For example, they may be hosting a website. In a secure network, public access servers are placed on a segment known as the demilitarized zone (DMZ). It is called the DMZ because fewer restrictions are placed on traffic entering or leaving public access segments than for private networks.



This firewall has three network interfaces. It serves to protect the private LAN from the internet and also to control access to web servers on a DMZ segment. The firewall ACL controlling access to the DMZ will allow more traffic than to the private LAN. For example, it may allow communication on TCP port 80—used for web page requests to the web servers to the DMZ—but not to the other LAN segment.

Identifying Firewalls

Firewalls often have two or three ports but may have more because some have built-in switches. In larger networks, they are usually rack mounted and may have the acronyms such as VPN (Virtual Private Network) or DMZ printed on them.



Icons used to represent firewalls on diagrams.

Proxy Servers

Proxy servers act as intermediary devices between client computers and servers on another network. They are similar to firewalls, and many of their functions overlap. However, a proxy server's purpose is to:

- Provide protection for internal hosts on a network by acting as the mediator for all transactions with the internet.
- Temporarily store frequently requested external data so that it can be provided to internal hosts without taxing the WAN link. (WAN connections are typically slower than LAN connections.)

Proxy Servers and Evidence

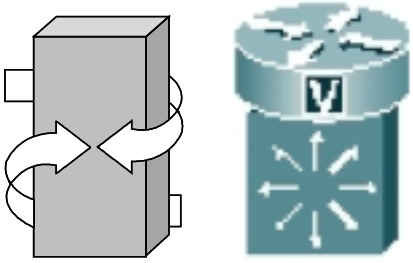
Proxy servers can record valuable application-layer information about the data they transfer for client devices. For example, a web proxy stores entire web pages and may record the IP address of the requesting computers for those pages.

Proxy servers are similar to firewalls that operate on the application layer, because they can filter traffic based on application-layer information through the use of ACLs.

Though a proxy server stores logs remotely, its cache of application-layer data is almost always kept locally on the device. The data cache is usually large. If you decide to collect this information, image the entire device.

Platforms, Placement and Recognition

Proxy servers can be hardware- or software-based, just like firewalls. They are often placed at the border between an internal network and the internet. Recognition techniques are also the same for firewalls.



Icons that represent proxy servers on topology diagrams. Proxy servers and firewall/proxy combinations are also labeled under the term gateway.

Lesson 5

Routers

All communication passing from one network to another must travel through a router. Consequently, router logs can provide evidence in an incident involving two or more networks. This lesson introduces students to routers and their importance to a network investigation.

OBJECTIVES *After completing this lesson, students will be able to:*
Describe a router
Explain the general purpose and function of a router
Recognize a router on a network diagram
Explain the types of data that might be obtained from a router

About Routers

A router enables data to travel from one network to another and may record certain aspects of this transfer. These logs may help identify who has connected to the network and determine a timeline associated with their activity. Although the data may not be as in-depth as firewall or IDS logs, it can still contain vital information.

Routers can be targets for attack. Obtaining the logs can help determine whether the router was directly compromised during an incident.

Routers and Data Storage

Routers contain mostly volatile data. While most routers have some form of non-volatile memory, such as CompactFlash (a form of FlashROM), the memory contains only limited information. Most of the data that you will gather is contained in normal RAM and will be lost if the device is turned off.

Traditional methods of powering the device down and obtaining a physical image cannot be performed on routers. Instead, you must access the router and obtain all data with the device on. This chart lists types of data storage on Cisco routers and the information they contain:

Storage Type	Description
ROM: Read-only memory	Bootstrap program and a limited OS used to boot the router
FlashROM: Flash read-only memory	The internetwork operating system (IOS) that runs the device
NVRAM: Non-volatile random access memory	The startup configuration
RAM: Random access memory	Running configuration, logs, protocol statistics and tables

Routers and Evidence

Routers contain different categories of information that may be useful in an investigation:

- **Configuration settings** provide details about how the device is configured to transfer data.
- **Protocol statistics and tables** include information about how certain network protocols are being used and limited protocol-specific records of network traffic.
- **ACLs** are used in a similar manner to firewalls.
- **Router logs** contain types of information that include ACL violations and configuration changes.

Configuration Settings

Two specific sources of configuration information may yield a large volume of information:

- The **startup configuration** is the set of parameters applied to the router when it is turned on or rebooted.
- The **running configuration** is the set of parameters the router currently uses. These include changes made to the configuration since it was booted.

Both sources of configuration information contain the same types of data, including:

- IP addresses and network interface status
- Routing and Network Address Translation (NAT) data
- Access lists on the device

This is a partial example of a Cisco router configuration, with a detailed explanation in the table that follows:

```
hostname Router01
logging buffered 16000 informational
enable secret 5 $1$WNSY$Tqc/H.PqR21hPbB09wgd60
username AdminBob password 7 05080F1C2243
!
interface Ethernet0/0
ip address 10.1.1.105 255.255.255.0
ip nat outside
interface Ethernet0/1
ip address 192.168.1.1 255.255.255.0
ip nat inside
!
ip nat inside source list 7 interface Ethernet0/0
overload
ip route 0.0.0.0 0.0.0.0 10.1.1.1
access-list 7
permit 192.168.1.0 0.0.0.255
privilege exec level 15 rlogin
line vty 0 4
transport input none
```

Configuration Output	Meaning
`hostname Router01`	Router's name: Router01
`logging buffered 16000 informational`	The router stores locally 16 KB of the most recent logs
`enable secret 5 $1$WNSY$Tqc/H. PqR21hPbB09wgd60`	Enable secret password provides encryption automatically using MD5 hash algorithm
`username AdminBob`	User account: AdminBob
`interface Ethernet0/0 ip address 10.1.1.105 255.255.255.0`	The interface Ethernet0/0 has an IP of 10.1.1.105
`ip nat outside`	The interface Ethernet0/0 is the outside interface for NAT translations
`ip route 0.0.0.0 0.0.0.0 10.1.1.1`	The IP address 10.1.1.1 is the default gateway for data leaving the router
`privilege exec level 15 rlogin`	rlogin has been made an administrative command
`line vty 0 4 transport input none`	Telnet management of the router is disabled

Protocol Statistics and Tables

While performing its job, a router must keep records of certain protocol-specific information so it can decide where to send traffic. These records may be of use. Here are some examples:

Table	Content
Routing tables	A router's primary function is choosing the best possible path for a transmission to reach its destination. Routing tables are lists of network addresses and methods for reaching those networks.
NAT tables	Tables of internal IP addresses and what external address they were acting under
TCP endpoint tables	A listing of TCP connections recognized by the router similar to the output of netstat in Windows and Linux

This is a line from a TCP endpoints table that shows a current connection to the router, with a detailed explanation in the table:

```
Local Address          Foreign Address        (state)
10.1.20.254.23         10.1.20.66.1103        ESTAB
```

Data	Meaning
`10.1.20.254`	IP address of router network interface
`23`	TCP port being used by the router for the connection
`10.1.20.66`	IP address of remote device connecting to the router
`1103`	TCP port being used by remote device for the connection
`ESTAB`	The connection is currently established (or active)

Access Control Lists

One function of the router is to provide policy-based access from one network to another. To accomplish this, routers make use of ACLs, which work similarly to firewall ACLs. Router ACLs use this specific information for making decisions:

- Source and destination IP address and port number
- Protocol type
- MAC address

This is an example of the use of a standard ACL in order to block all traffic except that from source 10.1.1.x.

```
interface Ethernet0/0
ip address 10.1.1.1 255.255.255.0
ip access-group 1 in
access-list 1 permit 10.1.1.0 0.0.0.255
```

Data	Meaning
`Interface Ethernet 0/0`	Applied to <interface >(inbound or outbound)
`ip address 10.1.1.1 255.255.255.0`	10.1.1.x network
`Ip access group 1 in`	The router examines all traffic it receives on the interface against the ACL
`Access-list 1 permit 10.1.1.0 0.0.0.255`	Block all traffic except that from source 10.1.1.x.

Note: Router ACLs are applied from the top down in the same manner as firewall ACLs. In this example, incoming traffic is compared first to the top line, which permits traffic from certain IP addresses, and then compared to the second line, which would deny everything that had not already been allowed through.

Log Locations

Routers can log different types of data and to send it to various locations. These are some common storage destinations for router logs:

- **Console/terminal logs** are instantly displayed on the screen for anyone who is currently managing the router. Console logging sends updates to any computer that is directly connected to the router, while Terminal logs are displayed for current remote administration sessions such as Telnet. When these logs have scrolled off the screen, they are lost.
- The **buffer** is a storage location of varying size in RAM, where logs may be temporarily kept. When the storage location fills up, the router will delete the oldest entries to make room for new entries. When the router is turned off, these logs are lost.
- Routers may send log entries to a remote logging server in syslog format.
- Routers may participate in a Simple Network Management Protocol (SNMP) managed network and be configured to send SNMP trap messages to the SNMP management device when certain events occur. SNMP traps enable an agent to notify the management station of significant events by way of an unsolicited SNMP message. These are data packages that are sent from the SNMP client to the server without being explicitly requested.

Note: Console/terminal logs are not useful. If buffer logs are present, they should be obtained directly from the router before it is turned off.

Log Categories

Router logs may include these types of information:

- Accounting statistics may include the source and destination IP addresses, number of packets, and bytes for normal traffic.
- ACL violations may be created for transmissions that violate an ACL rule. A router may log the source and destination IP address, port numbers, date, time, and protocol used.
- Router configuration changes, including the time and username associated with changes made to the router configuration.

This is a portion of a log retrieved from a Cisco router. It shows that a packet was rejected because it matched a deny entry in access list 7.

```
*Apr 27 01:47:40.704 UTC: %SEC-6-IPACCESSLOGS: list 7
denied 10.1.1.105 1 packet
```

Data	Meaning
`Apr 27 01:47:40.704 UTC`	Date and Universal Coordinated Time (UTC)
`%SEC-6-IPACCESSLOGS:`	This is an IP ACL violation event.
`list 7`	ACL number 7 was used to filter the traffic.
`denied`	The packet was rejected.
`10.1.1.105`	Source address of packet
`1 packet`	Number of packets rejected

This is a portion of a log retrieved from a Cisco router that shows that config mode was accessed remotely by the AdminBob account.

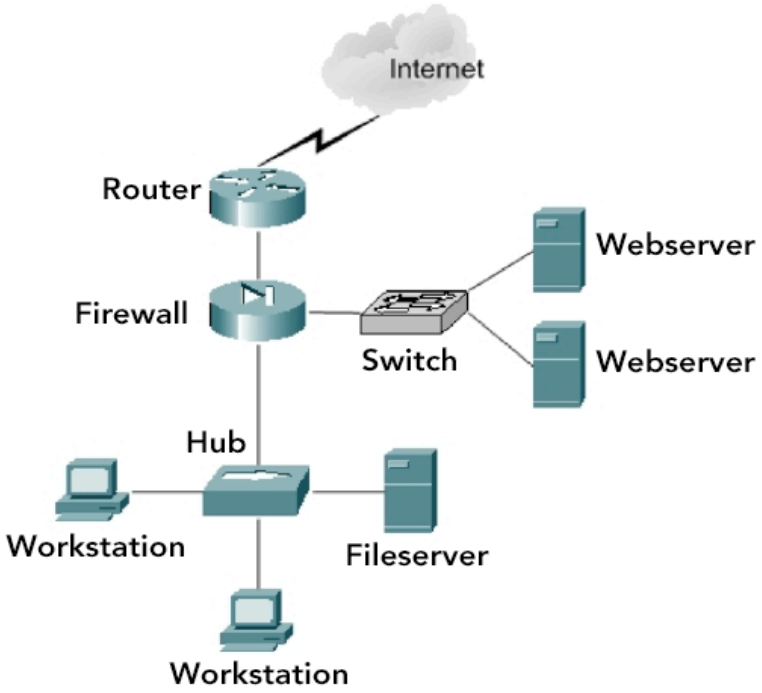
```
*Apr 27 02:30:01.158 UTC: %SYS-5-CONFIG_I: Configured
from console by AdminBob on vty0 (192.168.1.36)
```

Data	Meaning
`Apr 27 02:30:01.158`	Date and UTC time
`%SYS-5-CONFIG_I`	This is a configuration change
`Configured….by AdminBob`	The AdminBob account was used to access config mode
`on vty0`	Remote access was used over terminal vty0, probably Telnet
`192.168.1.36`	Source IP of remote access

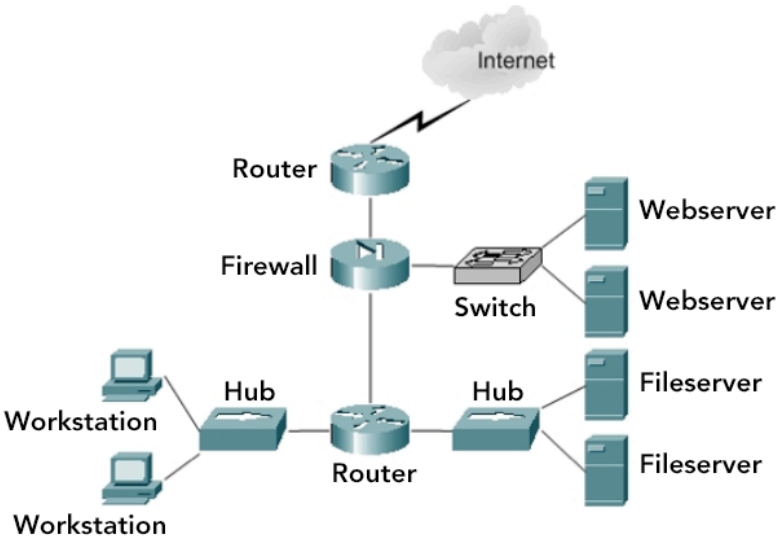
Locating and Identifying Routers

Because routers provide connectivity between networks, they connect an organization's entire network to the internet, usually through a line provided by an internet service provider (ISP). Routers are also used on internal networks to separate LAN segments and control the flow of data between them.

A router providing an ISP connection. The router is outside the firewall because it is responsible for directing traffic onto and across the internet and for connecting different types of media. In this case, internal LAN segments use Ethernet, most likely with CAT5 cabling. The connection from the ISP could be anything from a standard telephone wiring for digital subscriber line (DSL) to coaxial cable or fiber.



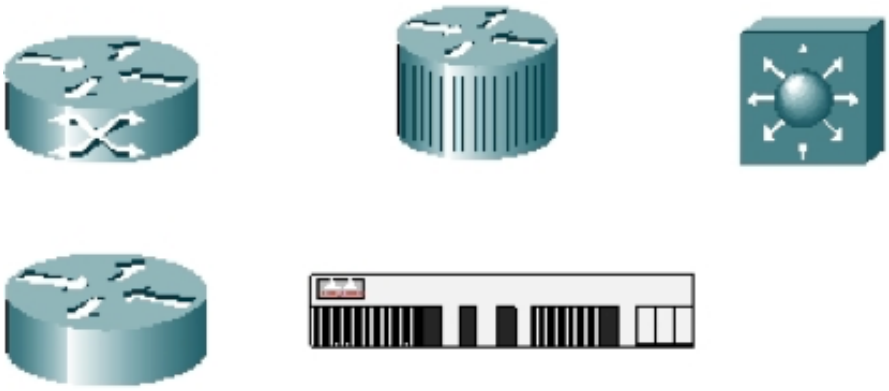
A router is used to place the file servers on their own segment. This separation allows an administrator to monitor the rate of traffic between workstations and servers and act as a security officer to control the traffic.



Identifying Routers

These factors relate to identifying routers:

- Routers always have at least two network interfaces.
- A router often includes network interfaces for more than one type of media, such as Ethernet and fiber.
- Some routers have expansion slots on the back for additional interfaces or additional Flash memory cards.
- Routers normally will be physically separating two or more networks.



Some of the many icons used to represent routers on a network.

Gathering Data on Cisco Routers

When you connect to a router, you can obtain several levels of access. These levels correspond to different tasks you can perform on the device.

Exec mode (User mode) is the mode you are in when you first access the router. This is a limited-access mode that only allows you to perform low-level functions, such as viewing basic router statistics, displaying the IOS version, or pinging another host. When in Exec mode, the prompt is the router's host name followed by the greater-than symbol:

```
Routername>
```

Privileged Exec mode is an administrative level of access that allows you to view and save the router's current configuration, debug errors, and enter various configuration modes. Access Privileged Exec mode by typing the command enable at the Exec mode prompt and entering the password. When in Privileged Exec mode, the prompt is the router's host name followed by the pound symbol:

```
Routername#
```

Configuration modes correspond to the components of a Cisco router, which allow you to configure those components. The prompt for the general configuration mode, also called the Global Configuration mode, is the router's host name, followed by "(config)" and the pound symbol:

```
Routername(config)#
```

The purpose of accessing a Cisco router during an investigation is to gather data. Never enter any of the available configuration modes. This will ensure that no accidental changes are made to the device. All necessary data collection can be completed from Privileged Exec mode.

Lesson 6

Sniffers and Intrusion-detection Systems

Network sniffers and intrusion detection systems (IDS) are devices designed to capture and analyze network traffic for specific patterns that indicate unauthorized activity. They can be valuable sources of detailed information for the analysis phase of a network incident. This lesson introduces students to sniffers and IDSs and their importance to a network investigation.

OBJECTIVES *After completing this lesson, students will be able to:*
Describe a sniffer
Describe an intrusion detection system
Explain the general purpose and function of sniffers and IDSs
Recognize a sniffer or IDS on a network diagram
State the associated layer(s) of the TCP/IP model for sniffers and IDSs
Explain the types of data that might be obtained from a sniffer or IDS

About Sniffers and Intrusion-detection Systems

A sniffer (also called a network analyzer or protocol analyzer) is an application used to record raw network traffic and present it in a readable fashion. This allows a network administrator or security officer to review network transmissions and search them for specific activities or trends. Sniffers only record traffic. They do not interpret meaning or intent.

An IDS is an application that provides automated analysis of network traffic for signs of unauthorized activity. IDSs record traffic and provide an explanation of what that type of traffic means.

How Does an IDS Work?

An IDS analyzes traffic by comparing it with known event signatures. A signature is a record of a particular undesirable network transmission, such as a Ping of Death attack. By comparing these signatures to captured traffic, an IDS can determine whether that traffic was part of an unauthorized action.

An IDS may also include an alerting mechanism which notifies an administrator when an attack is detected. Common techniques employed are screen messages, email, and pager alerts. An IDS also keeps logs of detected signature matches. This is the primary type of evidence it contains.

IDS Types

Two types of IDSs exist:

- A host-based IDS is software installed on a workstation or server for the purposes of detecting intrusion attempts against a specific device.
- A network IDS records and analyzes all network traffic that it detects, regardless of the source or destination.

Network IDS Components

A network IDS often consists of more than one machine with each device performing one of these functions:

- **Sensor:** IDS device that scans and records network traffic.
- **Console:** IDS device from which the system is configured and managed. A central console device often is used to manage multiple sensor devices.

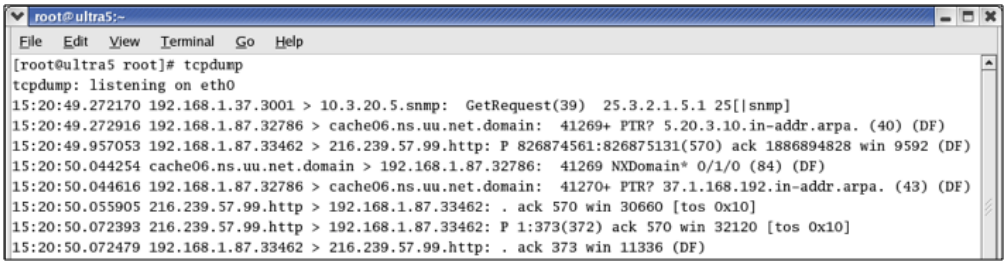
Sniffers and Evidence

Sniffers can capture raw network traffic. Their logs can contain the entire contents of each network transmission they record. How that raw traffic is displayed depends upon the type of interface the sniffer provides. Data displayed includes:

- Source and destination link-layer/Internet-layer addresses and port numbers
- A list of protocols in use from layers 2-4, and any specified parameters
- A listing of data transmitted in clear text

Sniffer Logs Example: tcpdump

A command-line packet analyzer, tcpdump, is a simple but powerful sniffer included by default in most Linux distributions. It records raw network traffic, but provides only a limited presentation.



A screen shot of tcpdump.

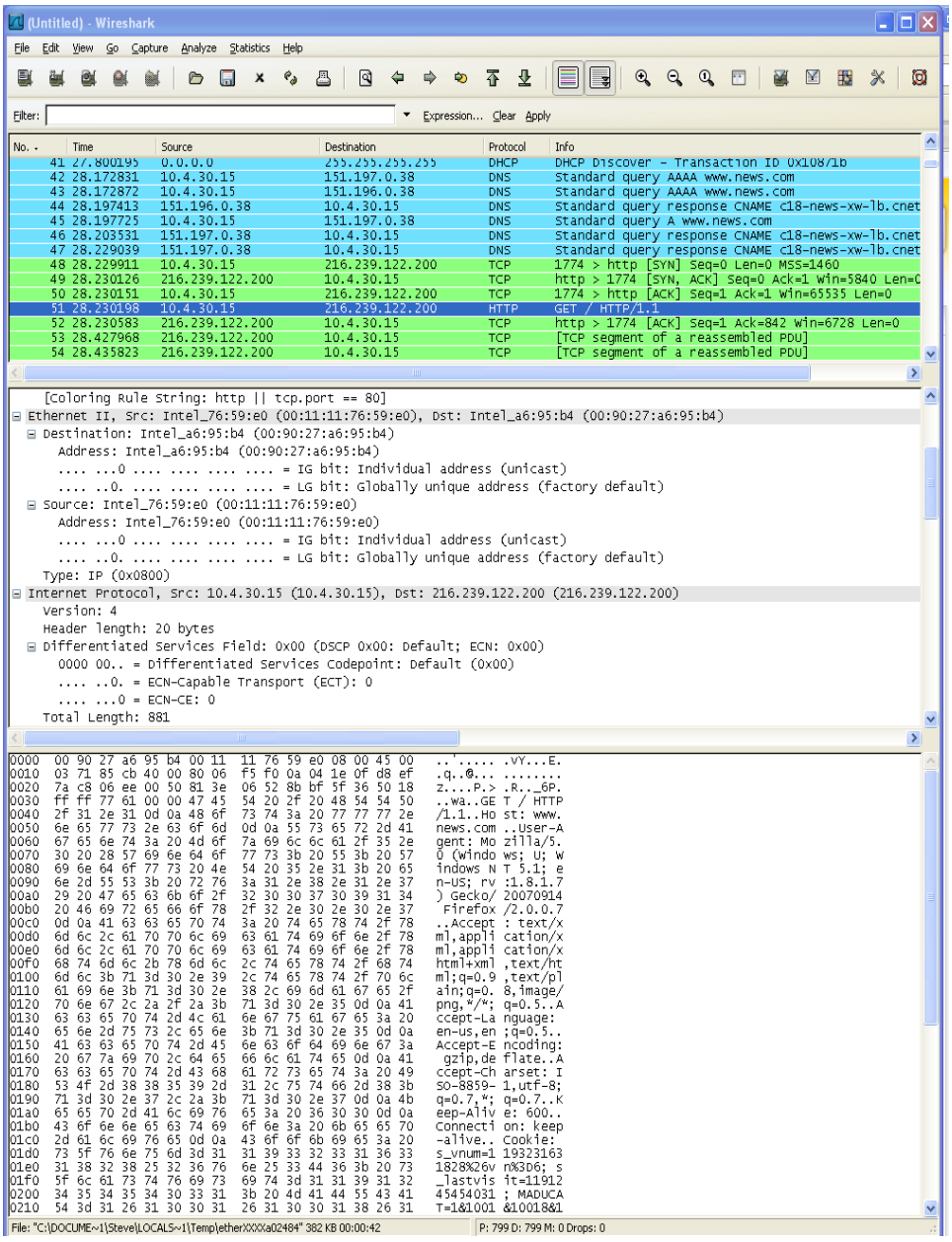
This table provides a description of the important sections of the first line.

Data	Description
15:20:49.272:70	Time in hours, minutes, seconds and milliseconds
192.168.1.37	Source IP address
3001	Source port number
10.3.20.5	Destination IP address
SNMP	Destination service (Port 25 interpreted as SNMP)
GetRequest(39) 25.3.2.1.5.1	Action taken by packet—an SNMP information request in this case

Sniffer Logs Example: Wireshark

Wireshark is an open-source GUI network protocol analyzer that captures network data packets and displays them for the user to browse interactively or from a previously saved capture file. Wireshark can read data from many other sniffers. Many display and capture filters are available to the user.

Three window views of the packets captured by Wireshark, from top: the summary window, the packet details window, and the packet in hex mode.



Intrusion-detection Systems and Evidence

An IDS provides three primary types of evidence:

- Signature files: Identify what is being scanned for by the IDS
- Alert logs: Record of alerts that occur when a transmission matches a signature that is being scanned for by the IDS
- General configuration: Should also be obtained to gain a complete picture of how the IDS is operating on a network

Signature File Example

Here is an example of a signature file entry for the software-based IDS called Snort. The entry is explained in more detail in the table that follows. Any traffic matching this signature would produce an alert message.

```
alert tcp $HOME_NET 16959 -> $EXTERNAL_NET any
(msg:"BACKDOOR subseven DEFCON8 2.1 access";
flow:from_server,established; content:"PWD";
classtype:trojan-activity; sid:107; rev:6;)
```

Data	Description
`alert`	A match will generate an IDS alert
`tcp`	Transmission is using TCP
`$HOME_NET`	Source is internal network
`16959`	Source port
`$EXTERNAL_NET`	Destination is external network
`Any`	Destination port
`msg: "BACKDOOR subseven DEFCON8 2.1 access"`	Text of the alert message
`flow:from_ server,established`	Alert triggered when: Message is server response to service request, and the connection is established
`content:"PWD"`	PWD is included in transmission contents
`classtype:trojan-activity`	Classification: Rule matches network trojan activity
`sid:107`	Number identifying the rule (100-1,000,000 identify rules that ship with Snort)
`rev:6`	Revision of the specific rule

IDS Logs

Some information in an IDS log is similar to that in sniffer logs, although the IDS also provides an interpretation of the transmission. Data in an IDS log includes:

- Source and destination IP address and port numbers
- Protocol in use
- Alarm triggered, including severity
- An explanation of the event or a link to an explanation

IDS Logs Example: Snort

Here is an example of a Snort IDS alert and an explanation of major components:

```
[**] [1:969:3] WEB-IIS webdav file lock attempt [**]
[Classification: access to a potentially vulnerable web
application] [Priority: 2] 07/17/03-16:21:42.035012
10.1.20.96:1032 -> 10.1.20.93:80 TCP TTL:64
TOS:0x0 ID:37937 IpLen:20 DgmLen:83 DF
***AP*** Seq: 0xB1AF2BE5 Ack: 0xE3581BC4 Win: 0x16D0
TcpLen: 32
[Xref => bugtraq 2736]
```

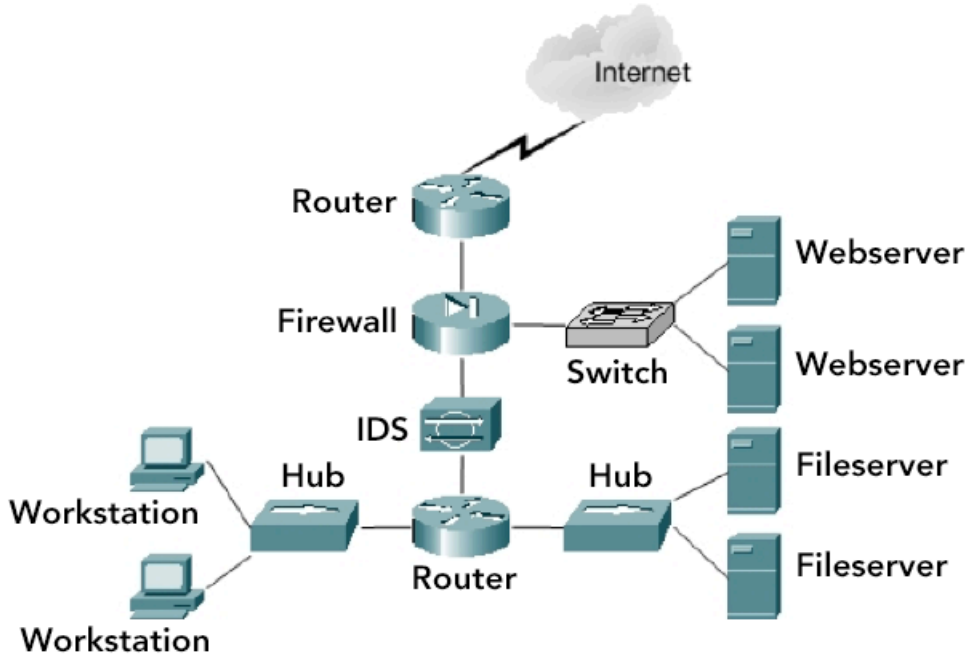
Data	Description
`WEB-IIS webdav file lock attempt`	Brief description of the event—in this case, an attacker is performing a denial-of-service (DoS) attack by repeatedly requesting a nonexistent file
`Classification: access to a potentially vulnerable web app`	Alarm category, which is manually defined in the security policy file
`Priority 2`	Denotes the level of severity for the alarm (lower numbers indicate a higher severity)
`07/17/03`	Date of alarm
`16:21:42.035012`	Time in hours, minutes, seconds and milliseconds
`10.1.20.96`	Source IP address
`1032`	Source port
`10.1.20.93`	Destination IP address
`80`	Destination port number
`TCP`	Transport protocol used
`TTL:64`	Time to live (TTL): The number of network devices (Internet layer and above) that a packet can pass through before it is dropped
`TOS:0x0….TcpLen:32`	Data from frame, packet, and segment headers
`Xref => bugtraq 2736`	Reference where more information on this event can be found (in this case, bugtraq 2736 points to information on the website securityfocus.com)

Locating and Identifying Sniffers and IDSs

A sniffer or IDS may take one of several placements, depending on its target and the setup of the network.

Network Placement: In-line

The first type of placement is in-line, which has two or more network interfaces and can be placed physically between two devices or networks.

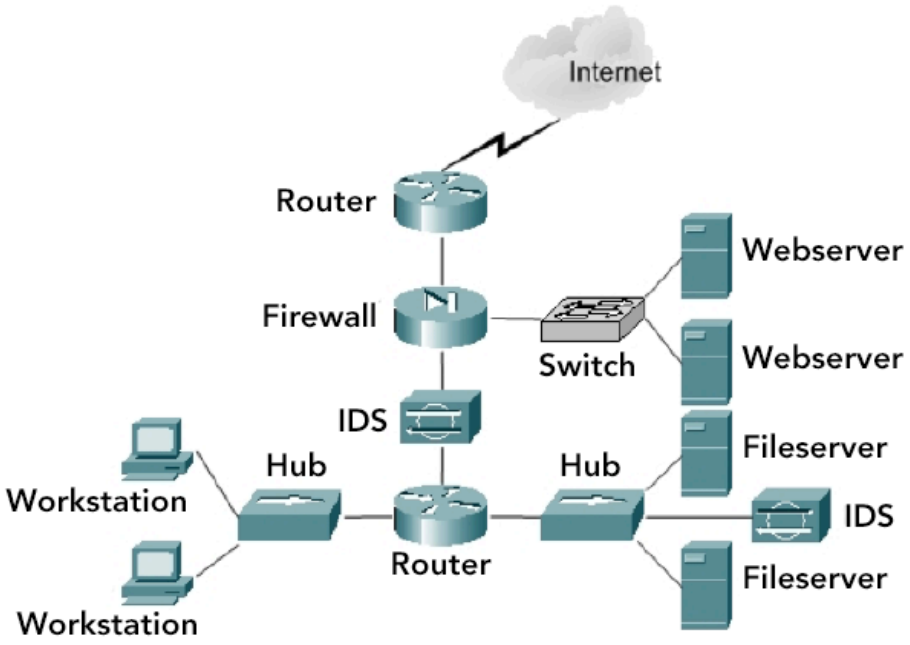


The topology of an in-line placement, where the IDS sits in front of the private LANs and behind the firewall and point-of-presence router. This is common placement for an IDS meant to scan all traffic between an organization's internal network and the internet.

Network Placement: Hub

An IDS may also be placed anywhere on a hub to monitor all devices attached to that hub.

A new IDS is attached to a hub with the two file servers. This IDS placement is able to detect all transmissions that occur between the file servers and the router to which the hub is connected.



Network Placement: Switch Monitoring Ports

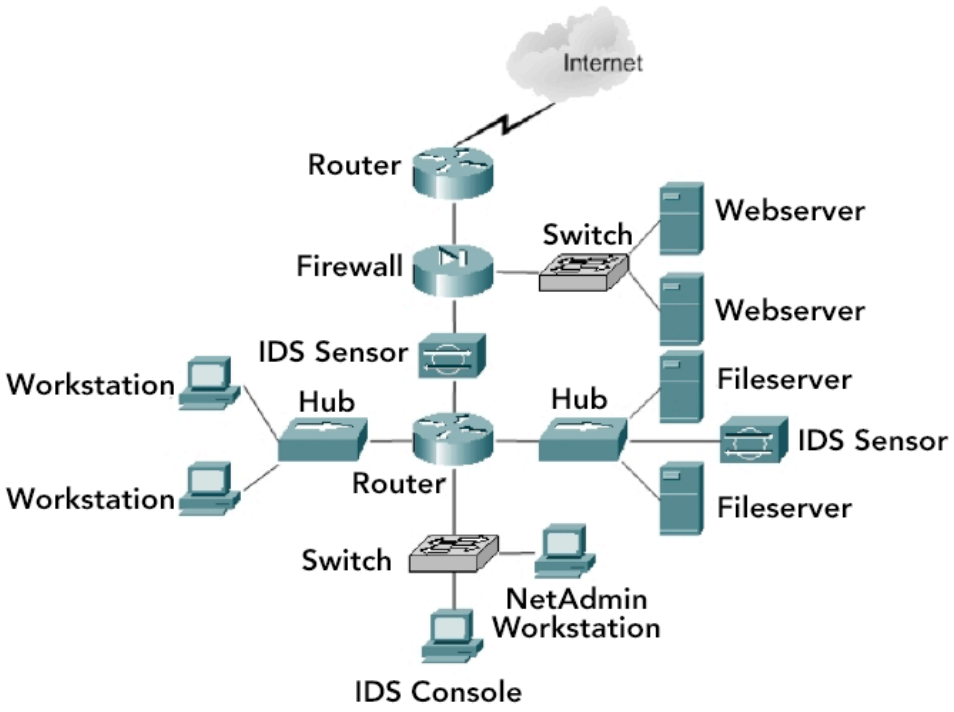
Unlike a hub, a switch does not forward every transmission that it receives out through each of its other ports. Because of this, an IDS or sniffer cannot detect communication between two other devices on a standard switch. Switch manufacturers solved this problem by offering a SPAN port on some models.

When a SPAN port is active on a switch, the switch will forward a copy of all transmissions out through that port, regardless of the source or destination MAC address. This enables an administrator to place a sniffer or IDS on that port and monitor all devices connected to the switch as if they were on a hub.

A diagram of this placement looks similar to the hub diagram, but a switch replaces the hub that connects the IDS and the file servers. If you see such a placement on a topology diagram, the IDS is most likely attached to a monitoring port.

Network Placement: Management Segments

On secure networks, the IDS console is often located on a separate LAN called a management segment. This is a LAN used only for network administration and security monitoring functions. It is kept separate for security reasons.



This diagram adds a management segment to the previous example. Notice that an IDS sensor is placed behind the firewall and another is on the collision domain with the file servers. The management segment contains the IDS console and the network administrator's workstation.

Identifying Sniffers and IDSs

Because many sniffers and IDSs are software based, it is not always possible to recognize them by sight. In fact, many administrators will place sniffers on a relatively inconspicuous machine. It might be necessary to reference the network documentation to locate the device.

Icons that are used to represent an IDS on a network diagram.

Lesson 7

Remote Logging

Larger environments often use a few devices as central storage locations for logs generated by all other network devices. These log servers can contain high volumes of network transmission records. This lesson introduces students to remote log servers and their importance to a network investigation.

OBJECTIVES *After completing this lesson, students will be able to:*
Explain the concept of remote logging
Explain the general purpose and function of remote log servers
Find remote log servers in a network
Explain the types of data that might be obtained from a remote log server

About Remote Logging

Networks use a variety of devices and OSs. When every device is storing its logs, auditing can become ineffective and impractical for several reasons:

- It is too time-consuming for administrators to access every device individually for regular log review.
- It is difficult to correlate activity between logs when they reside on different machines.
- The logs of a compromised device may also be compromised.

One solution for these problems is to have centralized, protected, and remote log storage. With this method, each device on a network sends its logs to a central log server, where they are stored in an organized and secure manner. This is sometimes called log aggregation.

Remote log servers store logs on hard drives and may also write copies of them to backup tapes or other media, such as a CD or DVD.

Syslog

The most common form of centralized, remote log storage is syslog. This is a simple application included in most Linux and UNIX distributions that accepts and stores log files produced both locally and remotely. Syslog can also be installed on Windows servers, but is not a default application.

Most of the settings for a Linux or Unix syslog server are found in a configuration file called /etc/syslog.conf. Reading this file tells you where the syslog program stores all logs that it receives.

Remote Logging and Evidence

Remote log servers can be used by almost any kind of device and contain a variety of logs. Device logs found on a remote log server include:

- Firewall and proxy server logs
- IDS and sniffer logs
- Router logs
- Linux/Unix file and e-mail server logs
- Individual host logs

Syslog.conf Example

A sample section from a syslog.conf file follows. This section tells where logs received from Router01 are stored, depending on their priority level. Note that this file only indicates that syslog is accepting logs for Router01 on the local6 facility. For reception and storage to take place, Router01 must also be configured to tag its outgoing logs with the same facility name (local6).

```
#Router01 General Troubleshooting
local6.debug
/var/log/router01_debug.log

#Router01 Minor Problems
local6.warning;local6.err
/var/log/router01_prob.log
```

Each line preceded by a pound (#) symbol is considered a comment and is there only to provide information to someone who reads the file. Two lines of configuration are presented; both are described in the following tables.

#Router01 General Troubleshooting	
`local6.debug`	This line describes the storage location for messages tagged with the local6 facility name and the debug priority level.
`/var/log/router01_debug.log`	Messages matching local6.debug will be sent to the file router01_debug.log in the path /var/log.

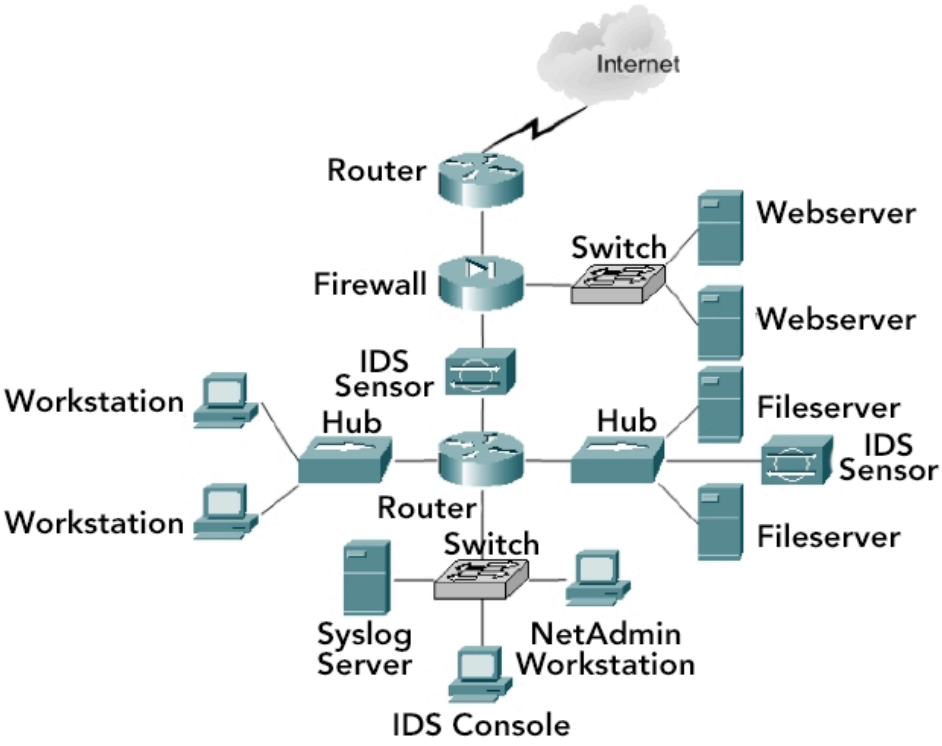
#Router01 Minor Problems	
`local6.warning; local6.err`	This line describes the storage location for messages tagged with the local6 facility name and the warning or error priority level.
`/var/log/router01_prob.log`	Messages matching local6.warning or local6.err will be sent to the file router01_prob.log in the path /var/log.

Locating and Identifying Remote Log Servers

Most remote log servers are software based, but some logging appliances can be found. With the increase in the importance of centralized logging, the amount of hardware-based logging appliances will increase. With software-based remote log servers, identifying the device requires referencing the organization's network documentation.

Network Placement

Remote log servers can be located anywhere on a network, as long as that network's routing/firewall infrastructure is configured to allow the other devices to send the remote log servers' data. They often will be placed on a management LAN segment for security purposes.



A network example with a syslog remote logging server added to the management segment.

Lesson 8

Logical Assessment

Key personnel need an effective way to identify devices that reside within a network. These people are usually network or system administrators who document, design and list the devices on their networks. Documentation illustrates where those devices are located and how communications take place within the network. This lesson shows students how to interpret network diagrams and documents to determine which devices should be considered for examination during an investigation.

OBJECTIVES *After completing this lesson, students will be able to:*
Describe the purpose of a network diagram
Interpret a network diagram to identify potential witness devices
Identify the devices that are illustrated in a network diagram

Network Documentation

Network documentation describes the physical and logical way a network is configured. Documentation illustrates what devices are within the network, where the devices are located, how they are connected, and how they communicate over the network.

Network administrators and system administrators who are responsible for the network usually keep this information documented. The documentation exists so the network can be configured, reconfigured, maintained, and repaired if trouble arises.

It is critical that this information is accurate and quickly obtainable. For example, when users report network problems, an administrator may first refer to the network documentation to start troubleshooting at either the source or destination of a problem. However, administrators who are familiar with their networks may not need to refer to the documentation.

Administrators must be aware of the entire network and the way it communicates. This is vital because administrators are often the primary contact for network problems within the organization. As such, their knowledge is essential to the organization, but can be a liability as well.

In some cases, only the system administrators have knowledge about the network. This is especially true in organizations with documentation that is out of date or even nonexistent.

Why Is Network Documentation Important?

Networks can be large, complex, and challenging to first responders, who must be able to read and interpret a network diagram and its associated devices. Often a first responder may not have time to access all the devices associated with the investigation, so it is important to be able to quickly identify which devices are most vital.

The responder should obtain and review a copy of the network documentation either before a search and seizure or at the scene. The more knowledge responders have, the better equipped they are to respond.

Current Network Diagrams and Documentation

Network diagrams and documentation must be current to be accurate and useful. A responder should inquire about when the documentation was last updated. Most network documentation may not be current because changes occur "on the fly" and updating is a low priority, if done at all.

The responder should also be sure to:

- Inquire about any hidden or new communications or networks.
- Perform a physical check for connections.
- Look for wireless symbols or use a wireless indicator.
- Examine small hubs, routers, switches, and loose cables and trace them for connections.
- Gather any backup/redundancy plans or documentation.

Device List

Make sure to obtain the device list that corresponds to the items on the network diagram and ensure that the information is current. A responder should know what the devices are and understand their purpose on the network. When in doubt about any of the items, ask someone familiar with the network.

For example, a device listed as Gateway on a network diagram may not necessarily be a firewall but an access point for routing, a Gateway machine, or just a name given by its classification. A responder can also search the internet for an item or visit the vendor's website for more information. Some items to look for include:

- Routers
- Switches
- Hubs
- Firewalls
- WAP
- Servers/gateway servers
- Workstations
- Wireless devices
- Network monitoring devices
- Access/network servers
- DMZ
- Dial-up connections
- VPN connections
- Communication links
- Data Centers/Network Operations Centers (NOC)
- Communication closets
- Patch panels

Considerations

Be mindful of these considerations:

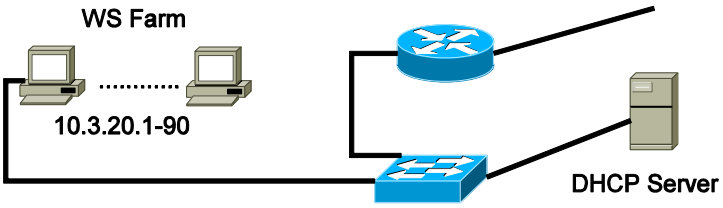
A system administrator who is the suspect may provide inaccurate information, intentionally or unintentionally. Consequently, the first responder's understanding of devices, the network, and the diagrams are important to the investigation. First responders must have a basic understanding of network documentation, its devices, and how they relate to one another within that network. This can help avoid some problems during information gathering.

A system administrator may state that the documentation exists only in his or her mind. He or she also may claim to be the sole owner of that information and the sole designer of the network. While this may be true, such a scenario usually only exists in need-to-know environments or with those who have job-security issues. Social engineering, investigative skills, and the rapport developed with the system administrator may determine what information the responder can obtain.

Protocol Effects: DHCP

DHCP affects the IP addressing scheme of the machines being investigated. DHCP maintains a pool of IP addresses and automatically assigns an address to a machine when one is requested. DHCP then regulates how long machines lease their IP addresses.

When an address refresh occurs, systems may be assigned new IP addresses. This can complicate and frustrate a responder's efforts. For example, assume a responder has focused efforts on locating a particular system based on an IP address. After a refresh, the address may no longer be assigned to the same system. In fact, the system may no longer be connected to the network or may have been assigned a new address. Another machine on the network may now be assigned the address the responder was looking for originally.



On network diagrams, look for devices showing ranges of IP addresses, instead of a single address. This will imply these addresses are distributed by a server or other device when prompted by one of the devices.

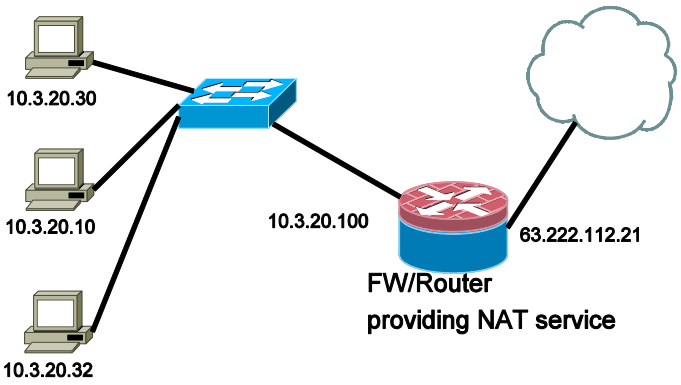
Protocol Effects: NAT/PAT

NAT and PAT also must be considered when locating evidence on network devices. NAT temporarily changes the non-routable internal IP address to an internet routable IP address.

If the network traffic indicates that the source of suspect data appears to come from a particular IP address, a NAT implementation on the suspect's network will hide the suspect's actual IP address behind a substitute address.

On a network diagram, routers that have this service running will usually have ranges of non-routable IP addresses showing on the inside interface, and a single IP address or a limited number of routable addresses showing on the outside interface.

PAT also changes the IP address, and assigns a unique port to the outgoing traffic. NAT and PAT can be running on the same network; however, each machine/internal IP address will use only one service at a time to communicate on the internet.



A network diagram showing routers with a NAT implementation service, which hides actual IP addresses behind a substitute address.

Logical Assessment Scenario 1

You are an agent responding to a situation that requires the collection of data from a witness device. On your request, an agent already investigating the situation provides you with these items:

- A brief description of the incident
- A list obtained from the system administrator of computers and other devices on the network and their functions
- A network diagram obtained from system administrator

Using this information, answer these questions:

1. List the possible in-line witness devices from which you can collect data of evidentiary value. What type of data do you expect to collect from each device?
2. List any other devices that might be of interest and explain why.
3. Based on the information provided, list any additional questions you would ask the network engineer or system administrator.

Event Description

This is a description of the incident provided by an investigator who has already been working the case:

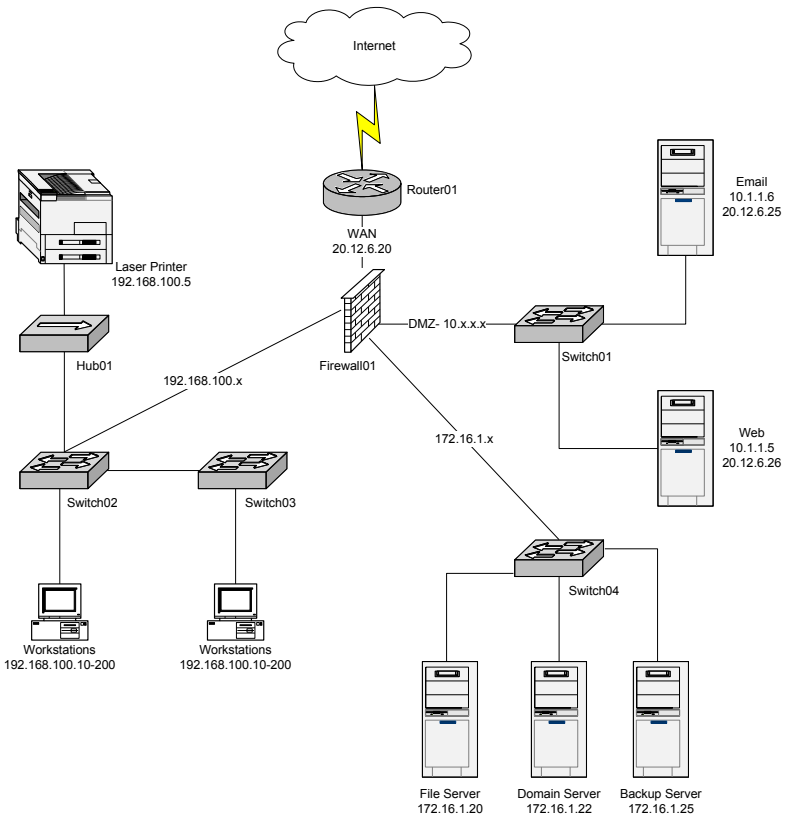
"This morning, several people reported that the base website had been defaced with anti-government material. The website is hosted on a server that is connected to the DMZ segment of the firewall. Reviewing the IIS logs, I located some suspicious Web requests from the IP address 214.3.152.67. It is suspected that the defacement originated from that IP address."

The base commander was notified. Per his instructions, a team will be sent to collect evidence related to the intrusion to determine what type of activity occurred. You are in charge of collecting witness device data and must decide from which device(s) to collect.

Network Documentation Device List

Device	Purpose	Make/Model
Router01	Border Router	Cisco 2600 Router
Switch01-04	Switches	Cisco 2950
Firewall01	Perimeter Firewall Internal: 192.168.100.1 DMZ: 10.1.1.1 WAN: 20.12.6.20 PAT: 20.12.6.20 Int. Servers: 172.16.1.1	Cisco PIX 515E
E-mail Server	Exchange Server Internal IP: 10.1.1.6 External IP: 20.12.6.25	Windows 2003 Server
Web Server	IIS Server Internal IP: 10.1.1.5 External IP: 20.12.6.26	Windows 2003 Server w/ IIS
File Server	172.16.1.20	Windows 2000 Server
Backup Server	172.16.1.25	Windows 2003 Server
Domain Server	172.168.1.22	Windows 2003 Server with Active Directory
Hub01	Printer Hub	3Com
Workstations 001-030	Workstations DHCP: 192.168.100.10-200	Windows 2000 Pro/XP

Network Diagram



Logical Assessment Scenario 2

You are an agent responding to a situation that requires collecting data from witness devices. On your request, an agent already investigating the situation provides you with these items (located on the following pages):

- A brief description of the incident
- A list of computers, devices and their functions, obtained from the system administrator
- A network diagram obtained from the system administrator

Using this information, answer the following questions:

1. List in-line witness devices from which to collect data of evidentiary value. What type of data do you expect to collect from each device?
2. Assume that Lt. Doe is suspected of attempting to transfer some of the unauthorized files off-site. From which devices would you want to collect data now? Why?
3. Based on what is provided, list any additional questions you would ask the network engineer or system administrator.

Event Description

This is the description of the incident, provided by an investigator who has already been working the case:

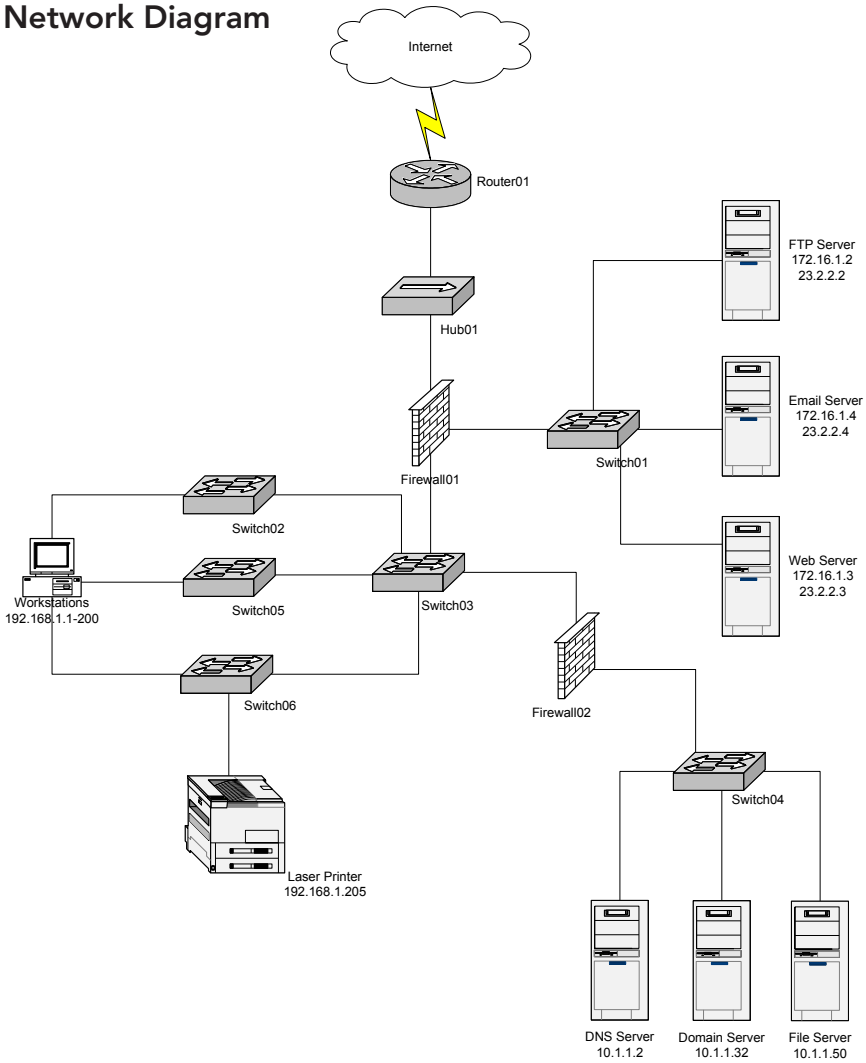
"Lt. J. Doe was accused of viewing unauthorized documents on an internal file server on the network. The documents are not classified, but Lt. Doe is not authorized to view or change them. Lt. Doe is suspected of viewing the documents because he referred to some of their specific content during conversations."

The base commander was notified. Per his instructions, a team will be sent to collect evidence from Lt. Doe's computer and witness devices to determine whether the suspected activity occurred. You are in charge of collecting witness device data and must decide from which device(s) to collect data.

Network Documentation Device List

Device	Purpose	Make/Model
Router01	Border Router	Cisco 2600 Router
Switch01-06	Switches	Cisco 2950
Firewall01	Perimeter Firewall	Cisco PIX 515E
File Server	10.1.1.50	Windows 2000 Server
DNS Server	10.1.1.2	Windows 2003 Server
Domain Server	10.1.1.32	Windows 2003 Server with Active Directory
FTP Server	172.16.1.2 NAT: 23.2.2.2	Windows 2000 Server
E-mail Server	172.16.1.4 NAT: 23.2.2.4	Windows 2003 Server with Exchange
Web Server	172.16.1.3 NAT: 23.2.2.3	RedHat Linux with Apache
Workstations 001-050	Workstations	Windows 2000 Pro/XP

Network Diagram



Logical Assessment Scenario 3

You are an agent responding to a situation that requires the collection of data from a witness device. On your request, an agent already investigating the situation provides you with these items (located on the following pages):

- A brief description of the incident
- A network diagram obtained from the system administrator

Using this information, answer the following questions:

1. List in-line witness devices from which to collect data of evidentiary value. What type of data do you expect to collect from each device? Identify each device.

2. List any other devices that might be of interest and explain why.

3. Based on what is provided, list any additional questions you would ask the network engineer or system administrator.

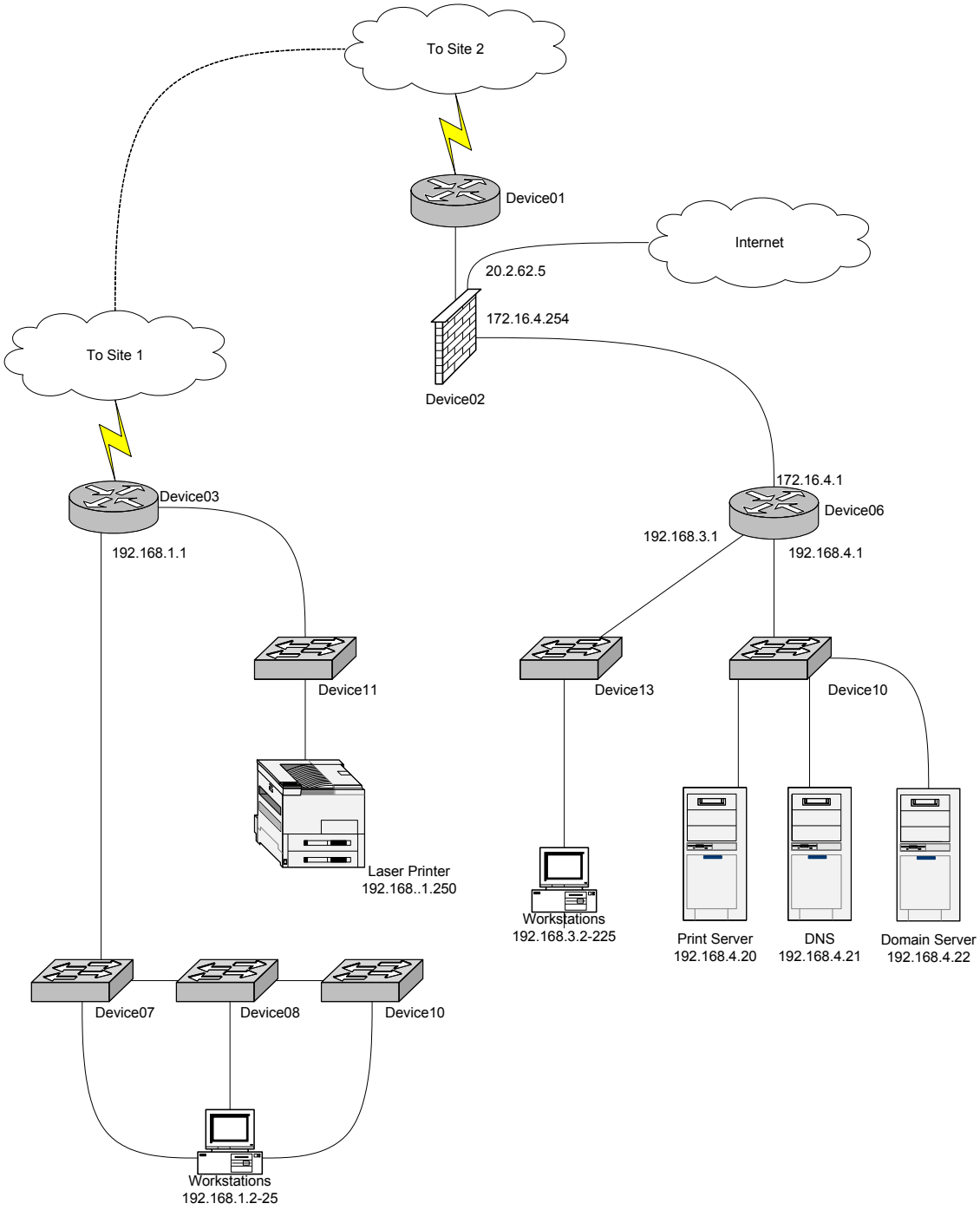
Event Description

This is the description of the incident, provided by an investigator who has been working the case:

"Lt. Lee Adama was accused by another person on base of viewing unauthorized pornography. This person shoulder-surfed Adama and saw him looking at the pictures in question. Lt. Adama works at the satellite office which is equipped with only some workstations and a printer."

The base commander was notified. Per his instructions, a team will be sent to collect evidence from Lt. Adama's computer and witness devices to determine whether this type of activity occurred. You are in charge of collecting witness device data and must decide from which device(s) to collect data.

Network Diagram



MODULE 3

Assessment and Sensor Placement

The proper configuration of network monitoring equipment is vital for a successful, legal capture of targeted data. Additionally, an investigator must understand and interpret the physical layout of the network to determine where to install the monitoring equipment without disrupting the network or revealing its presence. The investigator also should be skilled in configuring any existing network equipment to help the investigation.

This module teaches how to accurately place a network monitoring device on the network to gather the correct information while minimizing the chance of the device being detected.

OBJECTIVES *After completing this module, students will be able to:*
Recall the steps involved in network monitoring
Identify appropriate information from onsite personnel
Determine placement of a network monitoring workstation on the network
Explain how to install a physical tap

Lesson 1

Network Monitoring Methodology Overview

Network monitoring is the process by which computer network traffic can be captured, recorded and analyzed for specific content. The methodology presented here encompasses steps from preparation to reporting. Successful network monitoring is more than simply deploying a monitor and capturing network packets.

This lesson teaches the overall methodology of network monitoring and collection and identifies key information that can be gathered by questioning on-site personnel.

OBJECTIVES

After completing this lesson, students will be able to:

- Recall the steps in network monitoring
- Identify appropriate information from onsite personnel

Steps of Network Monitoring Methodology

This course focuses on the methods required to monitor network traffic for communication to and from a single host. The network monitoring methodology includes the following steps:

1. Preparation
2. Assessment
3. Deployment and collection
4. Retrieving captured data
5. Analysis and reporting considerations

These steps are described in greater detail in the following sections.

Monitoring Steps: Preparation

Before investigators begin monitoring a network, they must determine the requirements of the specific assignment. Requirements include:

- Legal restrictions
- Scope of search authority: What packets can be collected without violating expectations of privacy such as, the Electronic Communications Privacy Act (ECPA)?
- Target host information such as username, hostname, IP address
- Target communication types such as e-mail or chat traffic

Planning and Preparation

Before arriving at an incident response site, investigators must plan what equipment they may need. The makeup of network monitoring devices or sniffers is critical. Questions to ask in advance include:

- What monitoring software will be used?
- What hardware will be used for network monitors? Will responder's hardware be used, or hardware that is already on site and part of the functioning network?
- How will data be obtained from the sniffer?
- How will the sniffer be connected to the network?
- Will an out-of-band monitoring network be set up?
- How many network monitoring devices will be needed?
- How will the sniffer(s) be protected from malicious traffic?
- What equipment should be brought in the raid kit?

Monitoring Steps: Assessment

Once the requirements of the network monitoring assignment are established, investigators can identify the target host on the network and establish the proper monitoring position on that network. To do this, investigators must perform the following:

- Logical assessment
- Physical assessment
- Placement assessment

Using the results of these assessments, investigators can identify the optimal location for placement of the monitoring device.

Logical Assessment

Logical assessment involves obtaining any network topologies to get a rough estimate of where sniffers can be placed for an investigation. The network topologies may not exist or be severely outdated. Investigators can update the topology through interviews or by performing a physical/placement assessment.

Physical Assessment

Physical assessment includes tracing wire and cable to physical components on the network to create a wiring diagram. A wiring diagram shows the physical connections between devices on site and can help determine the accuracy of the logical assessment. Investigators can use several cable testing devices, such as a tone generator, to verify cable locations.

Placement Assessment

After a logical and physical assessment, investigators still may not have determined the ideal location for the monitoring device. Placement assessment involves taking sample traffic from a device or gathering configuration settings to determine the correct location on the network for the monitoring device. The sample traffic is typically called trap-and-trace data. Trap-and-trace data includes source and destination information. It may include traffic type. It does not include the actual packet data, or payload.

Monitoring Steps: Deployment and Collection

After establishing the monitoring location, connect the network monitoring device to the network. The goal when placing the physical device is to be as unobtrusive as possible and to disrupt the current network communication as little as possible. Ideally, the sniffer should be connected so that it is protected both physically and logically from adding unwanted communication to the network.

Setting the Capture Filter

Before starting to capture the network traffic, create and configure the capture filter. A capture filter imposes rules that determine which network traffic will be captured and which traffic will be dropped. Filters are used to comply with legal monitoring restrictions and to keep capture file sizes small to facilitate efficient analysis.

Activating the Monitoring Device

Once the network monitor is connected to the network and the proper filter is set, activate the capture software.

Verifying Capture

After the sniffer is activated, verify that it is capturing traffic from the target host, that the filter is operating properly, and that no unauthorized/unwanted communications are being collected.

Monitoring Steps: Retrieving Captured Data

After verifying that the sniffer has captured the proper data, hash and then remove the saved capture file(s) from the sniffer for analysis. Removal can be done by:

- Transmitting the captured data over the wire (optimally on a private or out-of-band network)
- Saving the captured data to an external storage device and removing it physically
- Removing the entire network monitoring device with the captured data

Monitoring Steps: Analysis and Reporting

After the data has been removed from the sniffer, analyze it for the required information. Analysis of the captured data can accomplish the following:

- Reconstructing specific events into a readable format
- Identifying the network addresses, computers or user accounts used in each event
- Identifying the date/time of each event
- Identifying the actions taken in each event

Information recovered during analysis is assembled into a forensic analysis report according to the standard operating procedures of the analyst's organization. This report should reflect the acquisition of the captured data and the relevant evidence found in the captures.

Interviewing Personnel

To effectively conduct investigative network monitoring, investigators need to know the technical and environmental information pertaining to the site. Information gathered from personnel interviews can greatly reduce the amount of time spent on the technical analysis of a network's logical and physical infrastructure.

Who Should Be Interviewed?

In addition to the leads developed from the complainant(s) and those in authority, people working in these positions should be able to provide assistance:

- Information systems security engineer and others responsible for the design and architecture of the computer and network security.
- Information systems security officer (ISSO) and others responsible for the day-to-day operation of the computer and network security.
- Physical security officer and others responsible for the physical security of the site. The ISSO may be involved with this function, especially if automated physical security systems are in use.
- System administrator, network administrator and others responsible for the day-to-day operation of computers, servers, clients, workstations, routers, switches, hubs, firewalls, intrusion detection systems, cable and wiring. Those performing these duties may be from the information management office.
- Configuration manager, release manager and others responsible for maintaining configuration and release of information that indicates when and why hardware and software changes were performed.
- Facilities manager and others responsible for the day-to-day maintenance of the physical plant.

Who Should Not Be Interviewed?

The personnel running the network may ultimately be the ones who need to be monitored. There is a thin line between obtaining help from personnel such as system administrators and avoiding tipping off subjects that they are being investigated. If the investigation precludes interviewing on-site personnel, then placement of the monitor becomes more difficult.

What Information Is Important?

Investigators are looking for any information that will further their process. First, determine who has site authority and coordinate with them to get their support. This person may also be able to provide and explain any

organizational charts, and to be a good source for other information, such as who performs the various roles in the organization.

Typical Questions

Have two goals in mind when entering a site to place a network monitor: finding the right placement for any network monitors and not tipping off the subject(s) being investigated. The capture files may not contain any relevant evidence if either one of these goals is not met. Interview questions should help accomplish both goals.

This is not an all-inclusive list of questions, but it should provide an idea of the types of questions to ask for a successful investigation.

Typical questions include:

- Do you have any network documentation, such as a network diagram?
- What is the type and speed of the network?
- Where is the subject's/victim's computer physically located on the network?
- What is the subject's/victim's IP address?
- Does the network use DHCP?
- What is the subject\victim's MAC address?
- Where is the default gateway?
- What type of activity needs to be monitored?
- What are the subject's normal work hours?
- To which areas does the subject have physical access?
- Does the subject have admin privileges?
- Is the subject computer-savvy?
- Did the subject sign a computer usage policy?
- Is there a banner waiving the right to privacy?
- Who is the point of contact (POC)?

Lesson 2

Monitor/Workstation Hardware

The target environment may affect the build of the monitoring workstation. The capabilities of the workstation should exceed the theoretical limits of the network to which it will be attached. If the workstation cannot keep pace with the speed of communications taking place on the network or is improperly configured, valuable data could be lost.

This lesson helps investigators determine the build of a monitoring workstation based on an understanding of common network specifications, hardware components and other factors.

OBJECTIVES *After completing this lesson, students will be able to:*
Describe the appropriate build of a monitoring workstation
Explain how the components affect the performance of the monitoring device

Network Monitoring Device Components

The capabilities of the target environment and the monitoring device hardware may vary. The network may be slowed by heavily loaded network components or poor construction. Once attached to the network, the monitoring hardware will operate under the same constraints and share system resources. Having a well-constructed monitoring system could be advantageous and help mitigate some of the constraints encountered on the target network.

Major Hardware Components of a Network Monitoring Device

Consider these major components when constructing a monitoring device:

- **Network interface:** The interface must be compatible with the target environment. As with all hardware on the monitoring device, make sure the hardware limits of the card match or exceed the theoretical limits of the target environment. After confirming compatibility, test the hardware beforehand to ensure it performs to expectations. When implementing an out-of-band network, two NICs will be required.

- **CPU:** The central processing unit must be able to handle the requests made by the monitoring workstation. The load on the processor may vary based on the OS and hardware.
- **Memory:** Depending on the software used to gather data, use a memory buffer to hold captured data until it is ready to be saved or analyzed. With a large buffer, the capture application will not have to save the data from the buffer as frequently, thus providing fewer opportunities for packet loss.
- **Motherboard:** The bus speed of the motherboard can be a concern in any system, especially on a network monitoring workstation. Most motherboards use the PCI bus to transfer network data to system resources. This can be a bottleneck for captured data depending on the clock rate and bus width. Various technologies are available to enhance bus speeds or to allow direct access to the Northbridge. Consider using a motherboard that can easily accommodate high-speed data transfers.
- **Storage:** Network monitoring may require large amounts of storage capacity. Factors affecting storage requirements are network load, filtering process and monitoring duration.

LAN Specification

This chart describes the theoretical maximum limits of specific target environments. These are just a few of the most common LAN environments found today.

Environment	Description
10baseT	• 10 megabits per second = 10,000,000 bits per second • 10,000,000 bits/8 bits = 1,250,000 bytes or 1.25 megabytes per second
100baseT	• 100 megabits per second = 100,000,000 bits per second • 100,000,000 bits/8 bits = 12,500,000 bytes or 12.5 megabytes per second
1000baseT	• 1,000 megabits per second = 1,000,000,000 bits per second • 1,000,000,000 bits/8 bits = 125,000,000 bytes or 125 megabytes per second
10GbaseT	• 10 gigabits per second = 10,000,000,000 bits per second • 10,000,000,000 bits/ 8 bits = 1,250,000,000 bytes or 1.25 gigabytes per second

Hardware

The choice of hardware depends on the type of investigation, equipment available and the type of target network. Laptops and desktops are the two main hardware choices to consider for monitoring workstations. Additionally, the possibility of using existing hardware on site or customized network monitoring devices called stealth devices (small form factor) will be covered.

Laptops

Using a laptop instead of a desktop for a network monitoring device has several advantages and disadvantages. Compact and portable, a laptop is easier to place in a congested network operations center (NOC), in a ceiling, or in another cramped area. Its built-in monitor and keyboard makes it easy to use and configure. However, a laptop usually has less power than a desktop and is harder to upgrade. When choosing a laptop, make sure it complies with hardware considerations.

Desktops

A desktop is less portable than a laptop and may need a monitor/keyboard/mouse for the initial setup on site. Even if the desktop is preset to automatically start capturing packets once booted, investigators will still need a screen to verify that the monitor is in the correct location and collecting the proper data. This limitation can be overcome by setting up a remote desktop capability that will allow investigators to configure the monitor over a network. The remote connection must be done over a private or out-of-band (OOB) network, not over the target network, where it might forewarn the suspect(s) of the monitoring operation. This also requires a second NIC and additional network connection for the monitoring system.

A desktop may seem like a poor choice. However, it can be easily upgraded on site to match the requirements of the target network. Adding additional NICs to a desktop to provide access to other networks such as an OOB network is easier. A desktop's components are usually more powerful, and the additional power can alleviate problems with dropped packets. Storage is a big consideration with a network monitor, especially when using a broad filter or no filter. When extra storage is needed, adding it to a desktop is much easier than with a laptop.

Existing Hardware

The problem with using a responder's laptop or desktop is that the equipment usually looks out of place. In certain investigations, the network monitor may be in a location that might tip off the subject(s) being monitored. Using existing hardware on site will make it more difficult for a subject to notice an ongoing investigation.

The use of on-site equipment has its own issues, including using software that is untrusted and equipment that is untested and may not work correctly with network monitoring software or equipment.

One method to alleviate those issues is to use any number of available live Linux distros.

Stealth PCs

Stealth PCs are compact computers that provide the power of a desktop but the compactness of a laptop. A monitor, screen and keyboard may be necessary for the initial configuration of these devices. Stealth PCs are usually designed to mount easily on a rack or placed in the ceiling. They provide capabilities like a remote power switch that will allow for easy powering of the device, especially if it is in a hard-to-reach area. The problems with upgrading a stealth PC are usually worse than with a laptop. The compact design usually does not allow any upgrading capability. Make sure the device comes with at least two built-in NICs so it has the capability to connect to a target and OOB network.

Software

The software used to gather data can vary depending on the OS on the network monitoring workstation. Most often, the monitoring system will be running either Windows or a variant of Linux. For this course, the network monitor will run Linux. The following are some of the commonly used applications for network monitoring with those systems.

Windows Network Monitoring

Common applications used for network monitoring with Windows:

- Wireshark
- tshark
- OmniPeek

- Windump
- Snort
- Putty

Linux Network Monitoring

Common applications used for network monitoring with Linux:

- Wireshark
- tcpdump
- Snort (sniffer and IDS)

Lesson 3

Network Tap Configuration

Sometimes investigators will not have the opportunity to connect their monitoring system to a switch to collect data. If this is the case, investigators can use a network tap and collect packets of data as they travel between the suspect system and a switch. Much like a SPAN port, a wiretap will send a copy of a packet of data to the monitoring system but will also allow the packet to be sent to its destination. Because of this, a physical network tap can operate without being detected. Physical network taps can be called wiretaps since they operate within the same principles and legal boundaries of traditional telecommunications wiretaps.

In this lesson, students learn the procedure for correctly connecting an Ethernet-based wiretap to a network to successfully acquire data packets. To do this, students connect a physical network tap between the monitoring system and another system on the network. Finally, students use Wireshark to gather packets from the suspect system.

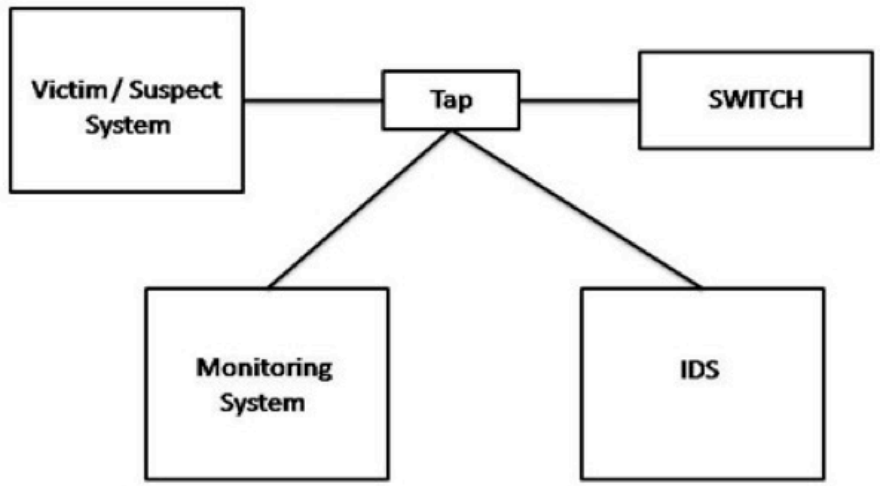
OBJECTIVES *After completing this lesson, students will be able to:*
Identify various types of taps
Explain the pros and cons of using taps
Install a physical network tap
Use a physical network tap to collect information being sent to and from a suspect system

Physical Network Taps

Network taps are used to monitor information that is flowing from one IT device to another. The single network cable used to connect the two devices is replaced by a pair of cables that allow the tap to be added to the network. The taps work in a passive and silent mode. They do not broadcast any packets themselves; they quietly make a copy of any packet they receive while also allowing the packet to travel to its destination. Most taps consist of three or more RJ-45 network ports. The first two are designated

for the source and destination devices: for example, a computer and a switch. The additional ports are designated for the monitor devices and may only include the inbound or outbound network traffic. Using a four-port TAP allows an administrator to run two separate monitoring devices at the same time. For instance, port 3 could be connected to an IDS while port 4 would be connected to a monitoring system running Wireshark.

Example of a four-port physical tap.



Why Use A Physical Tap?

In some cases, using a SPAN port is not a valid choice. For instance, a switch may be completely full or may not offer a SPAN port as an option. Additionally, physical access to the switch device may not be available when the monitoring is to occur.

Physical Tap Issues

One notable factor in using a tap is the break in network communications. To install a physical tap, a physical break is needed in the connection that exists between the suspect computer and the device to which it was originally connected. The break in communication can be as short as a few seconds, but it is required to install the tap. The original cable needs to be disconnected from the network and connected to one of the ports on the tap. Then an additional cable needs to make a connection between the second tap port and the network where the original cable was connected. At this point, the network connection will be re-established.

Fail Open vs. Fail Closed

Investigators need to know if the tap they are using will fail open or fail closed if power is removed. If a tap is a passive device and does not

regenerate the packet, then it will most likely exist in a fail open state if power is removed. A fail open device will continue to transmit the packets of data when its power source is removed. However, if a tap is an active device that does regenerate the packet before retransmitting it, it will most likely exist in a fail closed state when power is removed. A fail closed device will not transmit packets when power is removed. Even though computer networks are electrical circuits (where an open electrical circuit does not make a connection), terminology from the security field has been adopted (where an open door or lock allows things to pass through it).

Physical Network Tap Types: Simple Tap

A simple tap is an active or passive device that allows sniffing of traffic traveling along a data path. It is inserted in-line along the data path between two network devices. This type of tap produces output on two RJ-45 ports that mirror the output of each device. In other words, each tap port outputs only one side of the network stream. To capture full duplex network streams from this type of tap, two network interfaces must be implemented in the network monitor. This tap configuration at first glance might appear more limited and complex, but it allows much higher bandwidth monitoring.

Physical Network Tap Types: Aggregate Tap

An aggregate tap functions much like a simple tap, except it combines both sides of the monitored network conversation into the output RJ-45 monitoring port(s). This type of tap tends to be more expensive and have lower bandwidth capabilities but is usually more convenient to implement.

Physical Network Tap Types: Vampire Tap

A vampire tap is an older technology that was used to connect to a thick 10BASE-5 coaxial cable. A clamp containing a sharp probe was screwed onto the cable, piercing both the external and internal shielding. Once the probe made contact with the actual cable, the packets of data could be read and saved to a file for future analysis.

Physical Network Tap Types: Fiber Optic Tap

Fiber optic tap is a newer type of physical tap that, contrary to popular belief, proves it is possible to sniff information from a fiber optic cable. These devices actually bend the fiber optic cable to a point where the light beams can be detected through the protective outer covering. Some fiber optic taps have appropriate terminals to allow insertion of the tap in a traditional way and may offer the ability to provide output over traditional copper (RJ-45) connections. This capability is called media conversion.

Lesson 4

Physical Assessment

A logical assessment only gives an idea of how the network is set up. A physical assessment is necessary to verify that the logical assessment is correct. If no network documentation exists, then a physical assessment must be done from scratch and involves physically verifying how the network is set up by tracing wires and locating devices.

A physical assessment provides information that an investigator can use to select the proper location of a network monitor. This lesson demonstrates techniques to perform a successful physical assessment without network documentation.

OBJECTIVES *After completing this lesson, students will be able to:*
Conduct a physical assessment of the network site
Choose the best location for a network monitoring workstation after a physical assessment

Physical Site Examination

Examine the physical site to determine the physical data paths and their relationship to the overall physical environment. Understanding these relationships provides a basis for determining what is or is not physically possible on the network. A physical site examination includes the following tasks:

- Physically locate the target host
- Physically locate the device to which the target host is connected
- Physically locate devices that fall into the path of the investigation
- Verify the network documentation (if available)
- Determine the possible locations for a network monitor

Verifying the Network Configuration

When encountering an unfamiliar network, investigators need a starting point to verify the network setup and the actual location of network devices. Almost every network has a connection to the Internet or some external network. This external link is typically the best starting point to begin tracing wire.

Tracing wire is a technique used to determine how devices are physically connected to each other. If a wire cannot be traced because it is tightly bound to other cables or travels into a wall, other devices like a network tone generator can be used to determine its termination location. However, using some of these devices could require unplugging the cable and severing any existing connections, which could alert the suspect(s) to an investigator's presence.

Network Taps

To physically locate the target host, investigators must collect all identifying information regarding the device from reviewing the network documentation and interviewing the system administrator. Use this information and a physical assessment of the network to locate the device.

Network taps have at least three networking ports. The first port is usually designated for the suspect/victim machine. The second port would be connected to the original destination device, in most cases a switch located in the network closet. The third port would be used by the investigator's monitoring system.

Physically Locating the Nearest Device

Locate the hub or switch to which the target host is connected. This can be found by:

- Using the identifying information obtained during review of the network documentation.
- Recording the termination location for each network-capable cable connected to the machine. This could be an RJ45 or RJ11 socket on the nearest wall, a hub or switch, or some other device. If the cable terminates at a wall socket, record that socket's ID number and locate it on the patch panel that aggregates cables for that area of the facility.

Unable to Find Target Host

Sometimes a logical and physical assessment does not find the exact location where a target host is plugged into the network. The reasons could include not wanting to tip off the subject, inaccessibility or inability to find the exact network device.

If unable to find the right location for a network monitor after the logical and physical assessment, then use techniques of placement assessment as described in the next lesson to aid the search.

Lesson 5

Placement Assessment

Placement assessment involves looking at the logical data flowing through the network and configuration settings to determine the proper network monitor placement. Typically, performing a logical and physical assessment is not enough to determine the exact device to connect to the network monitor.

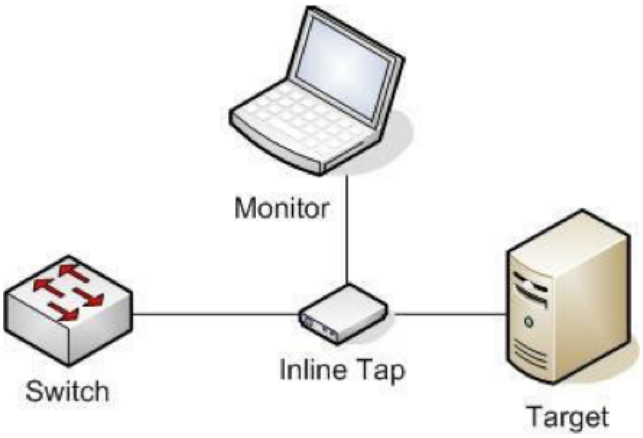
This lesson demonstrates several techniques to perform a successful placement assessment.

OBJECTIVES *After completing this lesson, students will be able to:*
Demonstrate how to interpret routing tables
Process and evaluate sample network traffic
Show which devices to ignore based on network traffic, configuration settings and scope of investigation

Placement Assessment Overview

To determine placement of the network monitoring workstation, first consider the logical and physical assessment. Selecting a location as close as possible to the target is preferable in most situations. This minimizes the amount of network traffic to the monitoring device and reduces the chance of missing traffic.

An example of a network placement setup.



The target could be outside the network (an outside attacker from the Internet), or getting close to the target is not possible because of fear of tipping off the target. In such cases, placing the network monitor next to the victim machine or some other location may be easier.

Legal Scope

Remember that the legal scope dictates where a network monitor is allowed to be placed and which network filters can be used. Consult legal counsel before placing a network monitor.

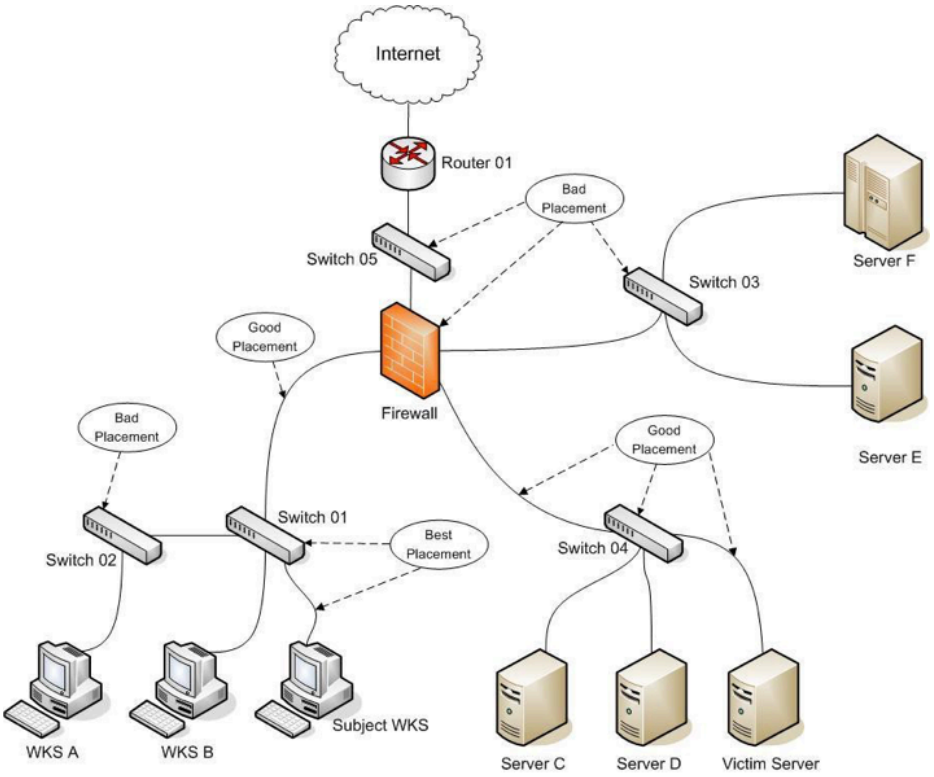
NOTE

The target machine is the one used by the subject that the investigator is attempting to monitor. The victim machine is the network resource the subject is accessing, and this access is the primary reason for the investigation.

Network Placement Example 1

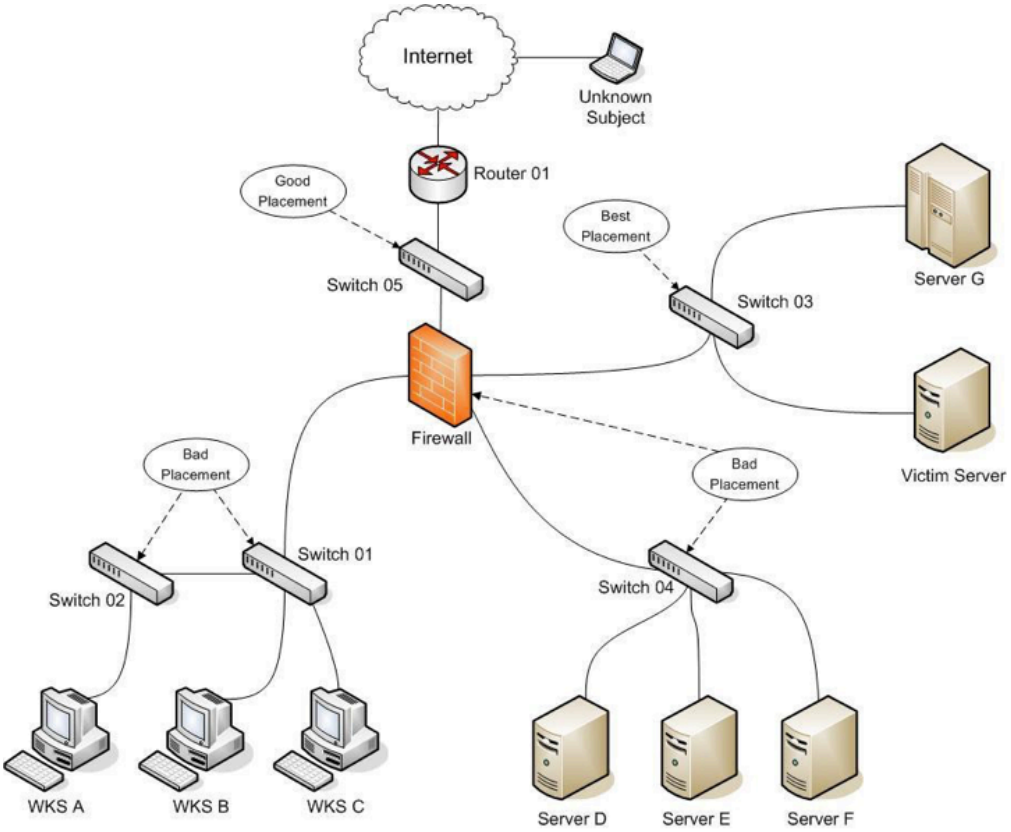
Examine this diagram. The subject is using a workstation on Switch01 and is performing actions of interest against the Victim Server on Switch04. Switch01, or an in-line tap between the subject's workstation and Switch01, is the best location because the network monitor will capture all traffic going to and from the subject's workstation.

Discuss the other locations in class. Also, if it is determined that the subject has access to a remote VPN account, how does this change the answer for the best placement?



Network Placement Example 2

Examine this diagram. The subject is now an unknown attacker coming from the Internet. The arrow pointing to the firewall indicating a bad location is placed in both examples to emphasize that an investigator should never try to place a network monitor directly into any firewall or router. In this example, both Switch05 and Switch03 are good locations to place a monitor. Discuss in class why Switch03 is the better of the two locations.



Location Determined

Once a general location is determined, a situation may exist in which the subject or victim may be connected to one or more devices. Imagine a large network where the subject is plugged into one of several switches set up for workstations. The logical assessment could not help the investigator because the network diagram did not identify the exact switch that each workstation is plugged into. The physical assessment was not a viable option because all workstation cabling is tied together and goes into the wall. Placing a cable tester at the subject's workstation is not desirable because it will likely tip off the subject to the investigation.

This is where placement assessment becomes important. Placement assessment involves any technique that consists of collecting logical data on the network to determine the proper location. This includes taking sample traffic, determining routing tables, and collecting configuration settings. Placement assessment will be continually practiced throughout the rest of the course.

Network Considerations

Many networks use switch devices because of their speed and reliability. Switch devices do not broadcast traffic to all ports and will only send data to a port that matches the destination MAC address (or broadcast traffic).

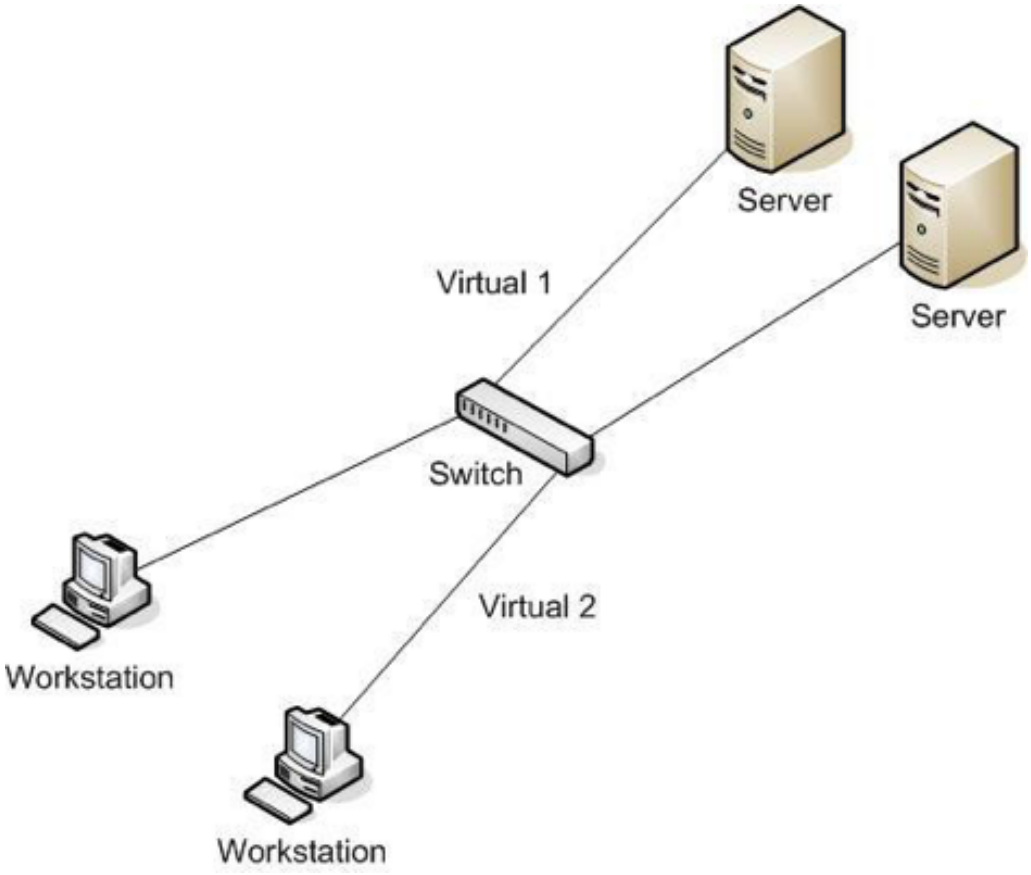
Switches usually have a console interface where network administrators can change the switch's behavior. The two main obstacles a responder will encounter is configuring a SPAN port and understanding VLANs.

VLAN

Virtual LAN (VLAN) is another way to divide a local area network into logical subgroups. VLAN uses software to connect a group of computers and devices together instead of manually moving cables and wiring. It can be used to combine workstations and other devices into a single group regardless of their physical location. The result improves traffic flow within the work group.

VLANs are used in LAN switches. Network changes and additions are quickly implemented with the VLAN software, making proprietary group solutions easy to create. VLANs operate at layers 1 and 2 of the TCP/IP model.

Two VLANs logically separated within the same network segment.



Response Considerations

Responders need to be aware of VLANs because they can affect the amount of traffic the sniffer can see. A VLAN is the virtual separation of traffic into different networks without the extra physical equipment. It is impossible to detect through physical assessment and must be done through logical and placement techniques.

MODULE 4

Capture

The deployment of a properly configured monitoring workstation, once the tools are installed, can vary greatly depending on the target environment and scope of the investigation, but the basic build and configuration should be flexible enough to work in virtually all scenarios with only minor configuration adjustments.

In this module, students learn to configure and execute the tools necessary to monitor live network traffic. Providing a methodology for activating and verifying a network monitoring device enables the exfiltration of data to other media or systems for analysis. This module focuses on a Debian-based Linux distribution to collect network traffic efficiently, effectively and reliably. The techniques taught may differ or not be applicable to other versions and distributions of Linux.

OBJECTIVES *After completing this module, students will be able to:*
Explain the difference between a trap-and-trace capture and full capture
Configure and use Wireshark
Describe tcpdump
Describe a data-retrieval methodology

Lesson 1

Trap and Trace vs. Full Capture

When conducting placement assessment, students examined logical network diagrams, physical network assessments, routing tables and witness device configurations to build a likely scenario of the best location(s) to place a monitoring device. One of the most useful procedures to hone and confirm correct monitor placement is via the collection of sample traffic. This procedure, called trap and trace, consists of collecting only packet header information to determine and/or confirm optimized monitor placement.

According to Title 18, United States Code § 3127:
"The term 'trap-and-trace device' means a device or process which captures the incoming electronic or other impulses which identify the originating number or other dialing, routing, addressing, and signaling information reasonably likely to identify the source of a wire or electronic communication, provided, however, that such information shall not include the contents of any communication."

This lesson introduces how data is transmitted over a computer network and the structure of that data. Students learn the difference between a trap-and-trace sample capture and a full-packet capture and conduct trap-and-trace captures using different tools. Additionally, students learn to interpret trap-and-trace data and how to use that information to hone and/or confirm optimal network monitor placement based on the desired traffic to be captured, investigative scope and monitoring efficiency.

OBJECTIVES *After completing this lesson, students will be able to:*
Interpret trap-and-trace data
Explain the difference between trap-and-trace capture and full-packet capture

Trap-and-trace Monitoring

A trap-and-trace monitoring session records data that summarizes the network traffic. The key characteristic of a trap-and-trace capture session is that although information about the packets is captured, the actual data contained within the packet is not saved in the capture file. Stated differently, only the packet header is saved, not the packet payload.

When conducting a trap-and-trace session, the aim is to collect only enough packet header data to identify the packets and information about them. The actual payload of the packet is not a concern. For IPv4 network traffic, this data resides in the first 96 bytes of the packet.

A trap-and-trace capture will consist of description information such as:

Item	Description
Packet number	Sequential number assigned to packets as they are captured
Timestamp	In seconds since the beginning of the capture session
Source IP address	IP address of the device sending the packet
Destination IP address	IP address of the device
Protocol	Protocol used to transfer the packet
Port	Port used by the particular protocol
Bytes	Number of bytes sent by the sender and receiver
Packets	Number of packets sent by the sender and receiver
Information	Text describing the function of the packet

Trap-and-trace Output

These lines of output are from a typical trap-and-trace monitoring session captured with the tcpdump application:

```
12:44:23.083818 IP 192.168.73.131.41705 > 192.168.73.2.53: 30864+ A?
www.google.com. (32)

12:44:23.083884 IP 192.168.73.131.41705 > 192.168.73.2.53: 13502+
AAAA? www.google.com. (32) 12:44:23.091996 IP 192.168.73.2.53 >
192.168.73.131.41705: 30864 6/0/0 A 74.125.26.106,[|domain]
```

```
12:44:23.094573 IP 192.168.73.2.53 > 192.168.73.131.41705: 13502
1/0/0AAAA[|domain]

12:44:23.094856 IP 192.168.73.131.33288 > 74.125.26.106.80: Flags
[S], seq 1314187429, win 14600, options [mss 1460,sackOK,TS val
153863 ecr 0,nop,wscale 6], length 0

12:44:23.114969 IP 74.125.26.106.80 > 192.168.73.131.33288: Flags
[S.], seq 324406470, ack 1314187430, win 64240, options [mss 1460],
length 0

12:44:23.115000 IP 192.168.73.131.33288 > 74.125.26.106.80: Flags
[.], ack 1, win 14600, length 0

12:44:23.115154 IP 192.168.73.131.33288 > 74.125.26.106.80: Flags
[P.], seq 1:542, ack 1, win 14600, length 541

12:44:23.116919 IP 74.125.26.106.80 > 192.168.73.131.33288: Flags
[.], ack 542, win 64240, length 0
```

These lines of output are from a typical trap-and-trace monitoring session captured with the application tshark (the command-line version of Wireshark):

```
4.009565 192.168.73.2 -> 192.168.73.131 DNS 133 Standard query
response 0x1c33 [Packet size limited during capture]

4.010257 192.168.73.2 -> 192.168.73.131 DNS 132 Standard query
response 0x5d6e [Packet size limited during capture]

4.025574 192.168.73.2 -> 192.168.73.131 DNS 89 Standard query
response 0xaa9c A 208.88.127.103

4.029512 192.168.73.2 -> 192.168.73.131 DNS 88 Standard query
response 0xa323 A 50.19.80.43

5.056726 192.168.73.131 -> 192.168.73.2 DNS 74 Standard query 0xa172
A www.google.com
```

```
5.056824 192.168.73.131 -> 192.168.73.2 DNS 74 Standard query 0x16f6
AAAA www.google.com

5.064841 192.168.73.2 -> 192.168.73.131 DNS 170 Standard query
response 0xa172 A 74.125.26.99[Packet size limited during capture]

5.065524 192.168.73.2 -> 192.168.73.131 DNS 102 Standard query
response 0x16f6 AAAA[Packet size limited during capture]

5.126609 192.168.73.131 -> 74.125.26.99 TCP 74 55497 > http [SYN]
Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=82050 TSecr=0 WS=64

5.145117 74.125.26.99 -> 192.168.73.131 TCP 60 http > 55497 [SYN,
ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
```

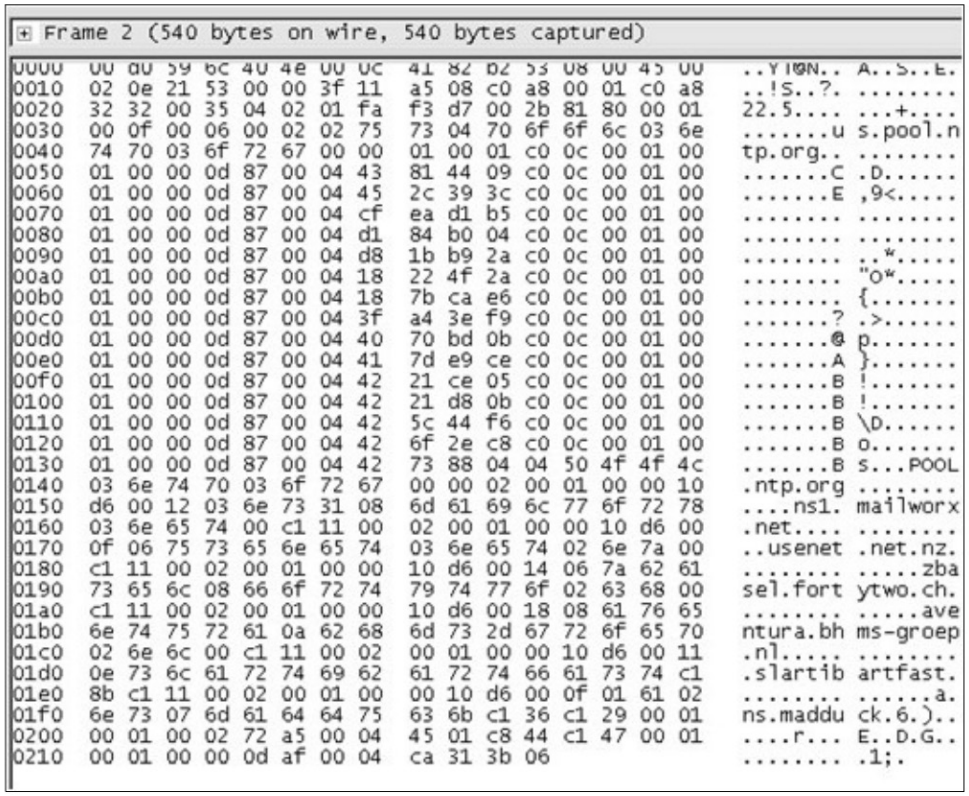
```
5.145143 192.168.73.131 -> 74.125.26.99 TCP 54 55497 >
http [ACK] Seq=1 Ack=1 Win=14600 Len=0
```

Identify and describe the crucial data fields in the preceding example trap-and-trace captures.

Full-content Monitoring

A full-content monitoring session contains all information collected in a trap-and-trace session, but it also captures the content, or payload, contained in the packets. It offers more information than the trap-and-trace monitoring sessions because it contains all the information passed between the two network devices. The results of a full-content session can then be searched and analyzed for pertinent information.

A sample packet from a full-content monitoring session using Wireshark. The header indicates that all 540 bytes of data transmitted were captured and are displayed. In many cases, the payload contains human-readable data.



Lesson 2

Capturing Data With Wireshark

Wireshark is a powerful, open-source protocol analyzer that can be used to view and capture network traffic. Wireshark can:

- Open a variety of binary log formats
- Act as a sniffer
- Translate or decode known protocols within a binary log to human-readable format
- Display highly detailed information on a frame-by-frame basis
- Search through a capture log for frames that match specific criteria
- Automatically reconstruct TCP sessions

Wireshark runs on various operating systems, including:

- Windows Server 2008/2012, 7/10
- Linux/Unix
- Mac OS X
- Solaris

The Wireshark package can be downloaded from www.wireshark.org.

In this lesson, students learn how to capture live network traffic with Wireshark, learn how to filter incoming and displayed network traffic, and understand the concept of a network stream.

OBJECTIVES *After completing this lesson, students will be able to:*
Configure Wireshark
Use Wireshark to capture network traffic
Implement capture filters and display filters
Use Wireshark to follow a TCP stream

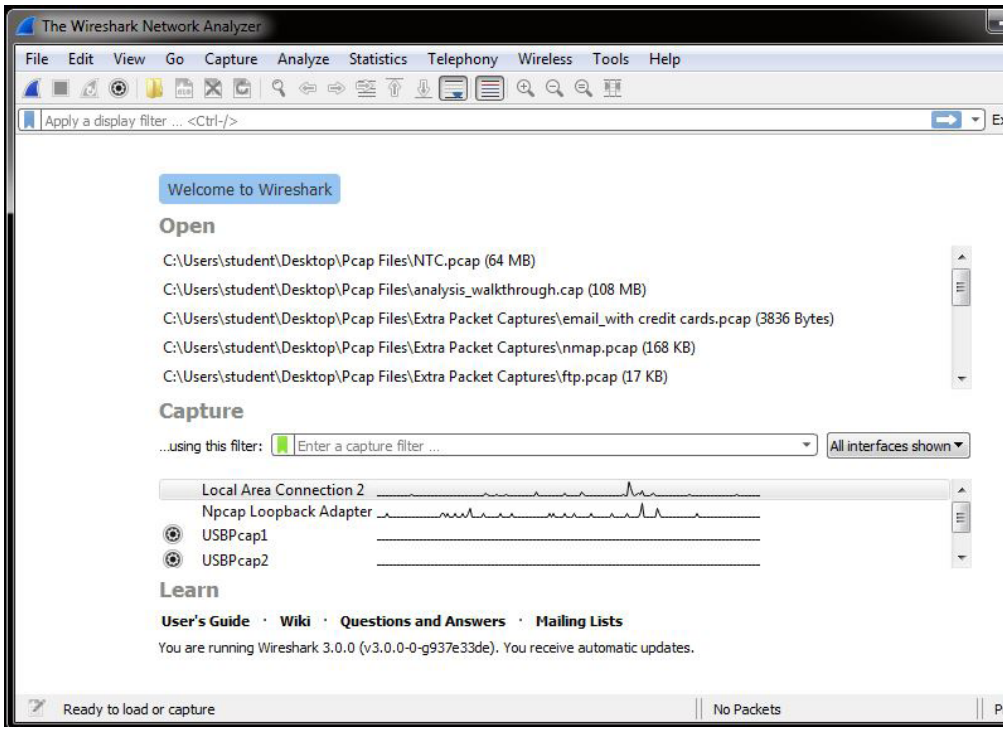
Wireshark Basics

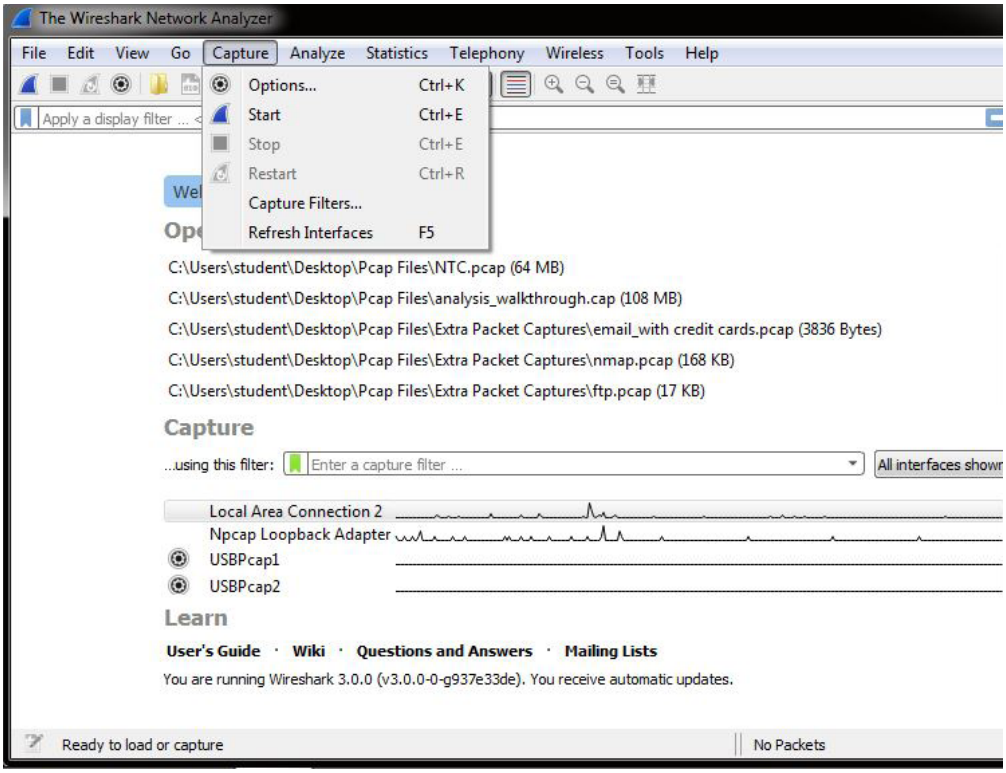
To efficiently analyze network communications with Wireshark, analysts must be familiar with Wireshark's Graphical User Interface (GUI). Wireshark is a capture and analysis tool with a plethora of options that customize the way it runs and displays data. The ability to understand the information each of the graphical elements is displaying and configure some of the more advanced options is paramount to getting the most out of it as a tool.

Wireshark is classified as a network packet analyzer. It attempts to capture network packets and display them in as much detail as possible in a way that is meaningful to the user. It can operate on live network traffic as well as previously captured traffic saved in a libpcap binary format. In short, Wireshark allows users to capture packets (or open previously captured packets) and display their content. Wireshark can also decode certain types of network streams and relate statistical information about particular network traffic it sees.

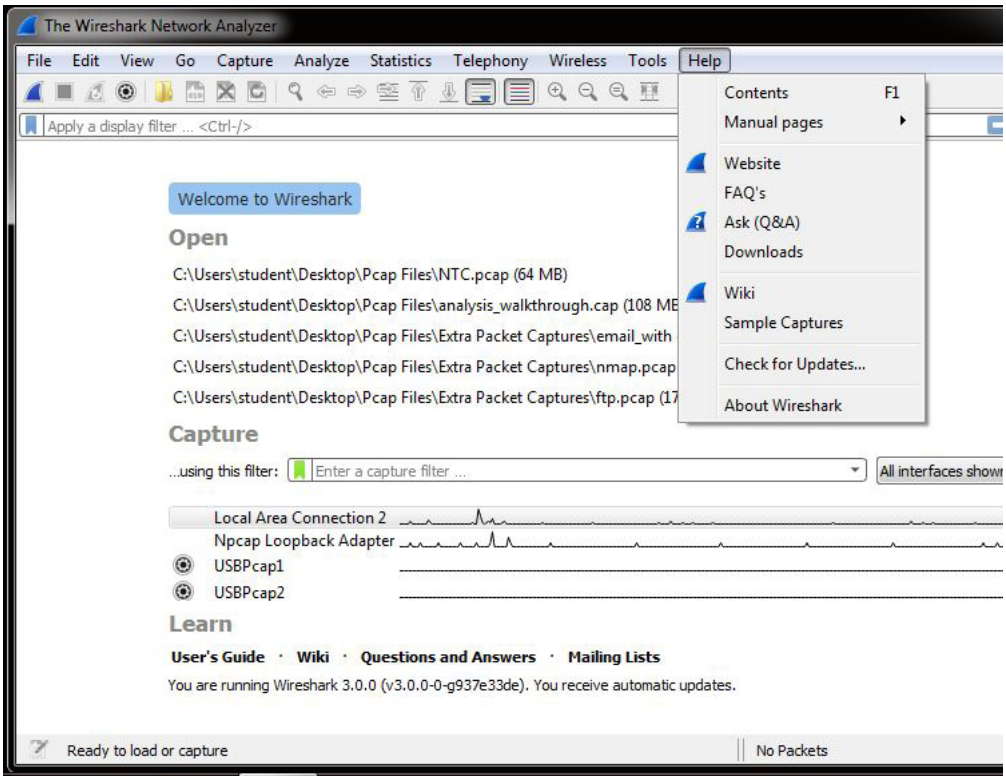
The most detailed and up-to-date information can be found on the Wireshark website at http://www.wireshark.org/docs/

Wireshark Start Page



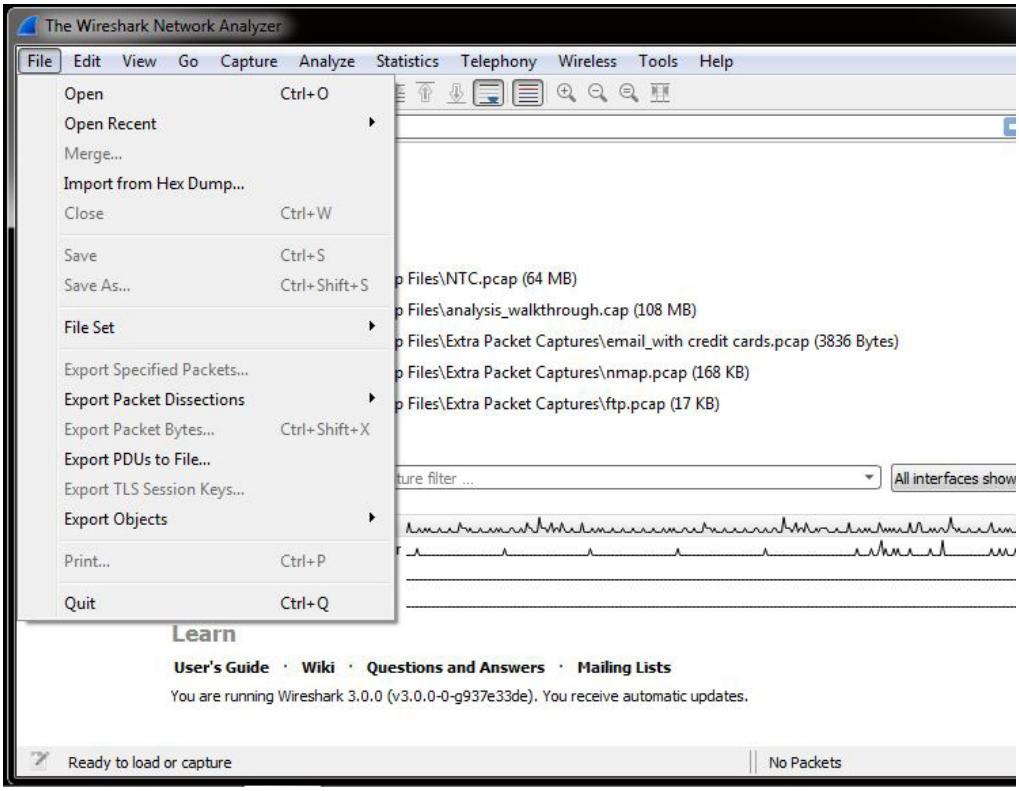


Capture allows users to select and configure an available network interface to begin a live network capture.

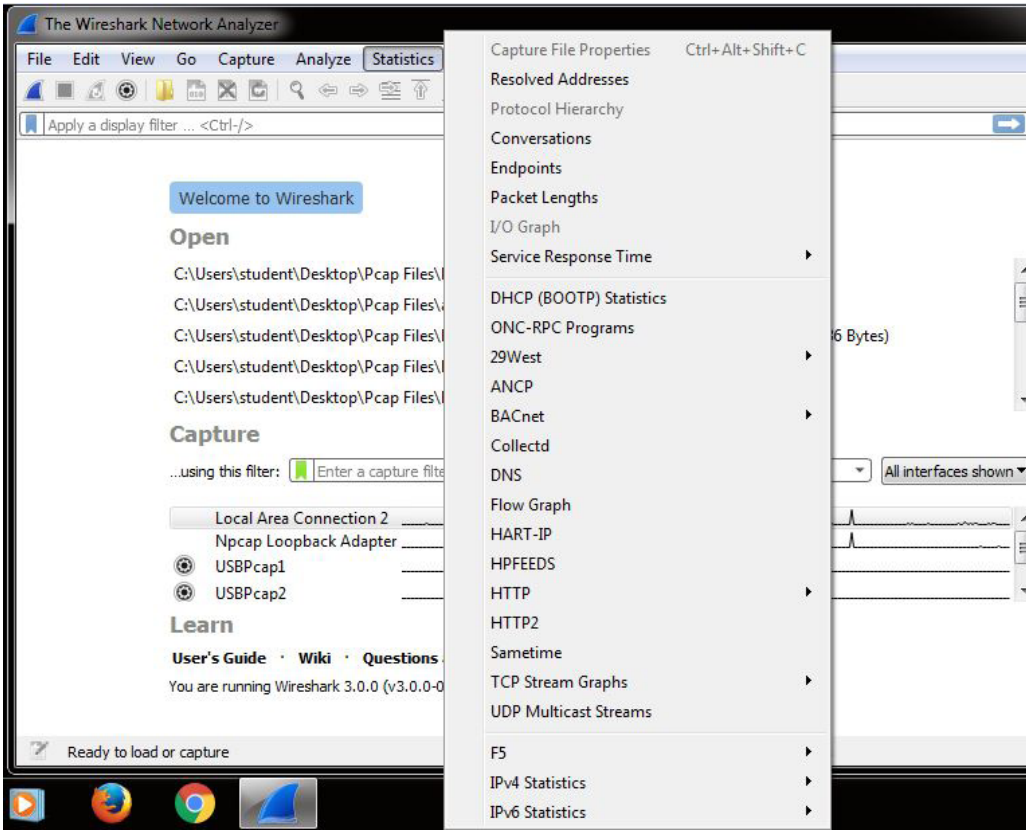


Help is available from the Help drop-down menu.

File provides options for loading previously created capture files, recently used files, and more.

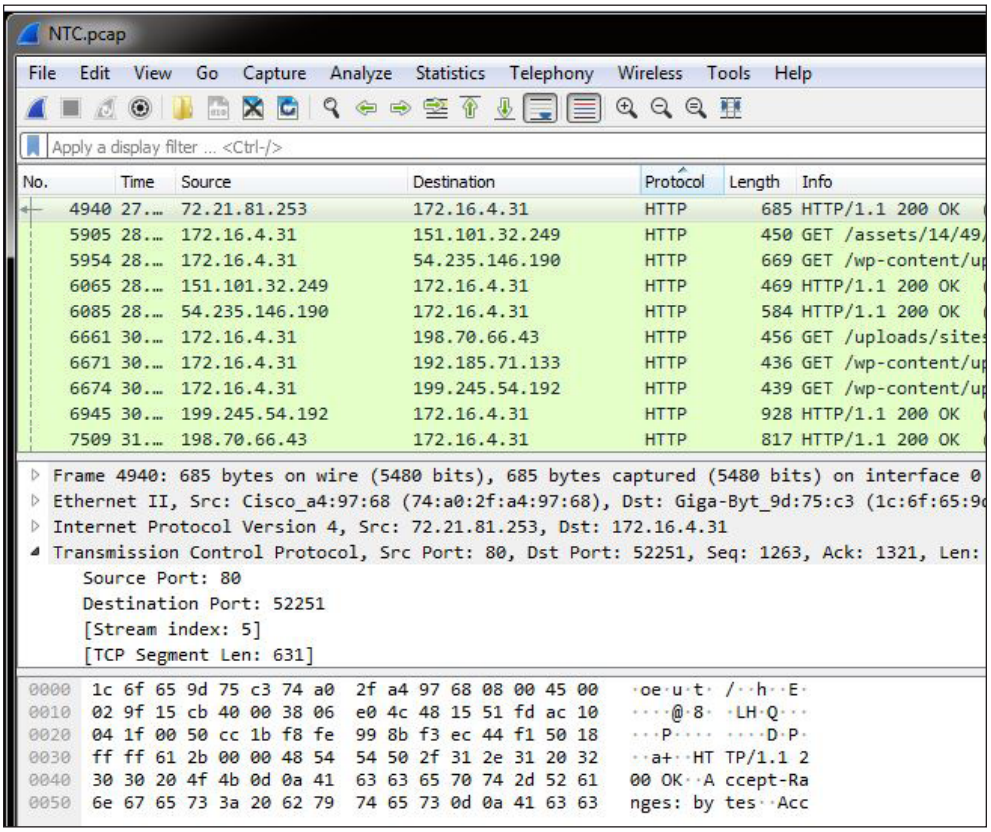


Wireshark provides a wide range of network statistics. These statistics range from general information about the loaded capture file (like the number of captured packets), to statistics about specific protocols.



Wireshark Display Panes

When working with captured network traffic in Wireshark, the data is displayed in various formats. These formats are presented in display panes. Wireshark displays three panes that present information about captured network traffic.



Wireshark's display panes.

Wireshark's top pane.

Wireshark's middle pane.

Wireshark's bottom pane.

The **top pane** shows captured frames and associated data applied by the application that captured the frames. This associated data includes the frame number, the frame arrival time, the frame origin and destination, the frame protocol, the frame length, and additional information gleaned from various parts of the header and payload.

Column headings can be clicked to reorder the display based on the information in the column. The example that follows shows unfiltered traffic, evidenced by contiguous frame numbers in the left column. Filtered views will exhibit gaps in these frame sequence numbers. Column content and column widths are customizable.

The Time column can be customized with different display settings, time index and time zone offsets through the main "view" menu of Wireshark.

Wireshark defaults to displaying time as a value relative to the beginning of the capture. While capturing, Wireshark captures time internally as UTC but displays time based on the time zone setting of the Wireshark user's computer to show local time relative to the time zone setting of the Wireshark user's computer. Wireshark automatically converts between computer time and time zone and UTC, so set the time and time zone correctly on both the capture and analysis workstations.

The **middle pane** shows the protocol details associated with the frame selected in the top pane. The pane presents frame and protocol information in a manner that closely follows the TCP/IP network layer model. However, the layers displayed start with the physical layer at the top and move down through the model with the application layer displayed at the bottom.

The display also presents relevant data contained within each protocol header. The display indicates that the HTTP application has been selected and is now highlighted. This selection determines what is highlighted and displayed in the bottom pane.

The information contained within each protocol section can be further expanded or "drilled" into by clicking the triangle to the left of the protocol area. The user can continue to move through the data and continue to drill down into specific information.

The **bottom pane** provides frame details associated with the protocol section selected in the middle pane. The bottom pane presents the content of the frame in hexadecimal and ASCII. Not all data can be represented in ASCII, such as when the data is encoded or encrypted. Also displayed is information associated with the client's Web browser settings - aka the User Agent String. Visible at the bottom is the status bar, which displays various statistical information about the capture. Due to the overhead of the Wireshark graphical display and packet analysis, dropped packets while capturing in Wireshark are common, particularly on systems with less than ideal hardware.

Wireshark Configuration

Wireshark is a feature-rich application that allows many aspects of the software to be customized. These customizations can affect the analysis, performance or network security when using Wireshark in any environment.

Wireshark Time Display Format

During analysis, correct dates and times are valuable and functional pieces of information that allow investigators to correlate network information with logs and alerts from other sources on the network. The information Wireshark and its command line utilities capture and analyze can be considered forensically sound data of evidentiary value. The data collected is captured as it travels across the transmission medium. Previously captured data is opened and viewed in the state in which it was captured.

The data collection workstation must have the correct date, time and time zone information.

This information from the collection machine is used in the time columns displayed in Wireshark. Without the correct information being set in the collection workstation, the time will not be correct in Wireshark. Therefore, make sure to set, verify and document these important configurations.

Some packets may carry timestamp information, such as http date and http last mod date. The presence of this information depends on the type of communication between two devices and the way the systems are configured to communicate. This information may or may not be relevant in certain situations or environments.

NOTE

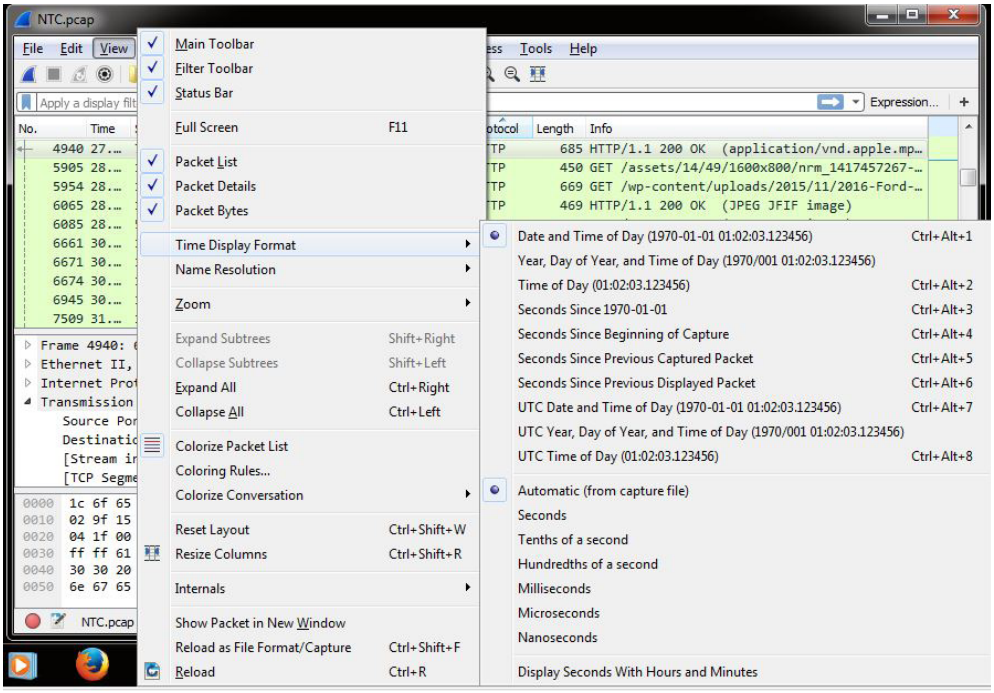
Because accurate time is so important in various situations, verify the authenticity and accuracy of the file when receiving a packet capture from outside sources. If the collection machine's time settings are incorrect, then the data in the file will be incorrect.

Wireshark provides several ways to customize the Time Column view. Each format has its own benefits. The Time Display Format can be changed at any time while viewing a capture file. When changing the Time Display Format, Wireshark will automatically reprocess all packets in the capture file and may take a little time depending on the size of the file being viewed.

To change the Time Display Format while viewing a capture file:

Navigate to View > Time Display Format| and then choose the desired option.

The available format options in Wireshark.



Time Zone Information

Because some capture files may have been captured in a location outside the current time zone, understanding how Wireshark deals with and converts time zone information is important.

NOTE

Verify the time zone in which the data was captured as well as the application used. Some applications store UTC values, and other may store local time values. When this happens, Wireshark must first convert the local time to UTC time before updating the display to the local time zone on the machine on which it is being viewed.

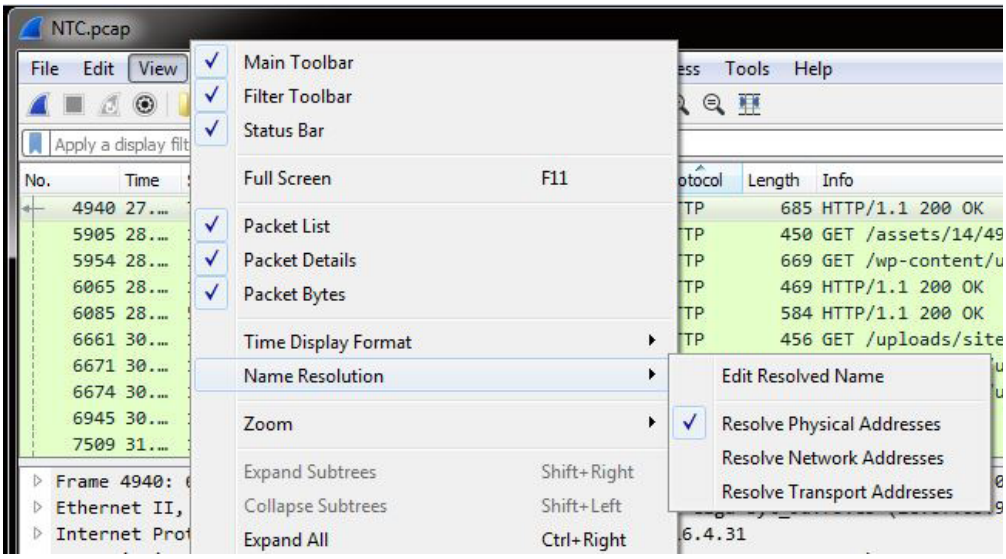
Wireshark captures use the libpcap format, which saves the packet arrival time of the packet as UTC values, in contrast to local time values. This means Wireshark will convert the UTC time in the capture file to local time automatically when opening a capture file. This is done by using the local system time zone offset configured in the system.

This is important to be aware of because a packet that displays 11 a.m. in New York would display that same packet as 8 a.m. in Los Angeles because of the automatic time zone conversion.

Wireshark Name Resolution

Wireshark can resolve the names at the MAC, Network and Transport layers. This can be useful in certain situations by making it easier to analyze and/or troubleshoot the data. In other situations, allowing any type of name resolution may not be wise.

Name resolution interprets a number value to something that is more descriptive and user-friendly. This feature can be useful in many circumstances. It can help quickly identify the DNS name of an IP address, identify the manufacturer of an Ethernet device, or resolve common ports to services.



An example of Wireshark's name resolution options. Options with a check are enabled. To enable or disable name resolution, navigate to View | Name Resolution | and then choose the desired option. Once a name resolution option is enabled or disabled, the capture file may need to be reloaded (refreshed). Do this by navigating to View | Reload (Ctrl + R).

Side Effects of Name Resolution

Name resolution has important negative side effects. Wireshark does not save any resolution information in the packet capture file. The resolution takes place when the file is opened or during the capture process, which will consume resources. Here are key points about name resolution.

- **Name resolution may fail:** Resolution depends on the system that is viewing or capturing the data. Therefore, the local system might be unable to successfully resolve the requested information.
- **Resolution information is not stored in the capture file:** Wireshark name resolution is meant to assist the user. This information is not stored in the capture file, so the data displayed by Wireshark could appear to be different on different machines or even at different times on the same machine. This depends on the success or failure of the resolution information requested at the time in which the data is being viewed. Therefore, DNS could resolve to one DNS name from and external source, and then when it is looked at in the internal system, it could resolve to a different DNS name if using the internal DNS server. Also, the DNS information could be changed or updated from the time the data was captured to the time the data was examined.

- **The resolution may be captured as part of the capture:** Because data could be required to travel across the network to be resolved, those requests also could be captured. For example, if DNS resolution is turned on during capture, the DNS lookup request from and response to the monitor system might be captured as well.
- **DNS resolution is cached by Wireshark:** If the DNS information is updated while viewing the data, then Wireshark would not be aware of this and could display outdated information.

MAC Layer Resolution

MAC Layer Name Resolution in Wireshark attempts to resolve a MAC address to a more user-friendly name. Wireshark attempts to resolve the MAC address in three ways in a particular order.

- **ARP Name Resolution:** Wireshark polls the OS to try to convert the MAC address to an IP address.
- **Ethernet Codes:** If ARP resolution fails, then Wireshark will try to convert the MAC address to a friendly name by searching a file called "ethers." This file could reside in two places. A personal ethers file would be stored in the same directory as the preferences files. This allows the user to create a custom file that contains friendly names of known devices on the network. If the personal ethers lookup fails, then Wireshark looks for global ethers that could be found in the Wireshark install directory.
- **Vendor Codes:** Lastly, Wireshark attempts to resolve the MAC address to a vendor by searching the "manuf" file, stored in the Wireshark install directory, by matching the first three bytes of the address. If a match is found, the first three bytes are replaced with the vendor name, followed by the last three bytes of hex such as: Intel_3E:C8:5C

Network Layer Resolution

The Network layer resolution attempts to resolve an IP address to a hostname. This can help identify sources and destinations quickly within the capture file.

Wireshark uses a certain order for name resolution. This order can be used to make identifying resources much easier and allows for the names to be customized to the user's liking.

- **First Attempt (Wireshark hosts file):** Wireshark will search the hosts file located in the local Wireshark directory associated with the active user profile. This file can be customized.
- **Second Attempt (system hosts file):** If the first lookup fails, it will then look to the system hosts file for a match.
- **Third Attempt (DNS Lookup):** If the previous local lookups fail, then Wireshark will use the OS and request a DNS lookup.

Potential Negative Side Effects

If network layer resolution is enabled while the DNS server is down, then Wireshark may significantly slow down while it waits for the DNS timeouts. If this feature is enabled while capturing, then the request from the capture machine could get captured in the file. The capture machine itself could be detected if it is sending requests out on the network. Take cautionary steps if a capture needs to be conducted covertly.

Transport Layer Resolution

The Transport layer resolution deals with resolving port numbers to protocol, such as port 25 > SMTP, 21 > FTP, etc. This is done by polling the "services" file in the Wireshark installation directory.

It can be useful to quickly identify the services associated with a port. However, Wireshark only matches the port number with the service from the services file.

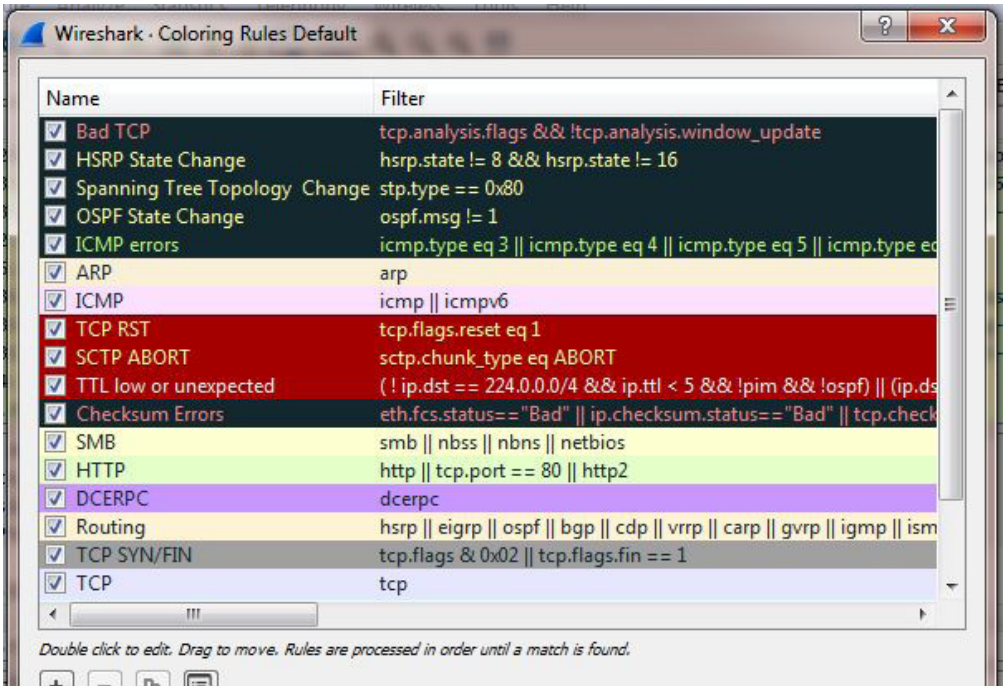
Wireshark does not perform analysis on the packet to determine the type of traffic being transmitted across the wire. This is important because potentially harmful traffic can be configured to use any of these ports in an attempt to hide data transfers taking place over the network or through network security devices. For example, if malicious TCP traffic is configured to tunnel over port 53, Wireshark will simply report it as DNS traffic; it does not identify the traffic as TCP data.

Wireshark Colorization Rules

When analyzing data, displaying important information in an easily recognizable way is helpful.

Coloring rules can be customized to help highlight particular packets of interest by defining the background and font color for them. This is an easy way to make packets stand out.

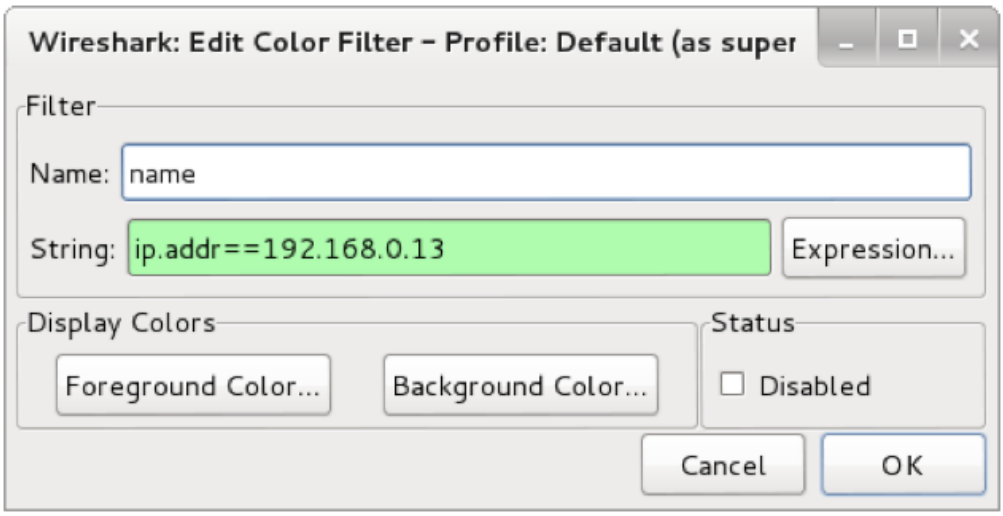
This window in Wireshark allows users to edit the current color rules. The rules are processed from the top down. This makes it possible to have multiple rules match a single packet, but only the first matching rule takes effect.



The current coloring rule set in Wireshark can be activated or deactivated quickly by selecting the "coloring packet list" circled icon on the toolbar.

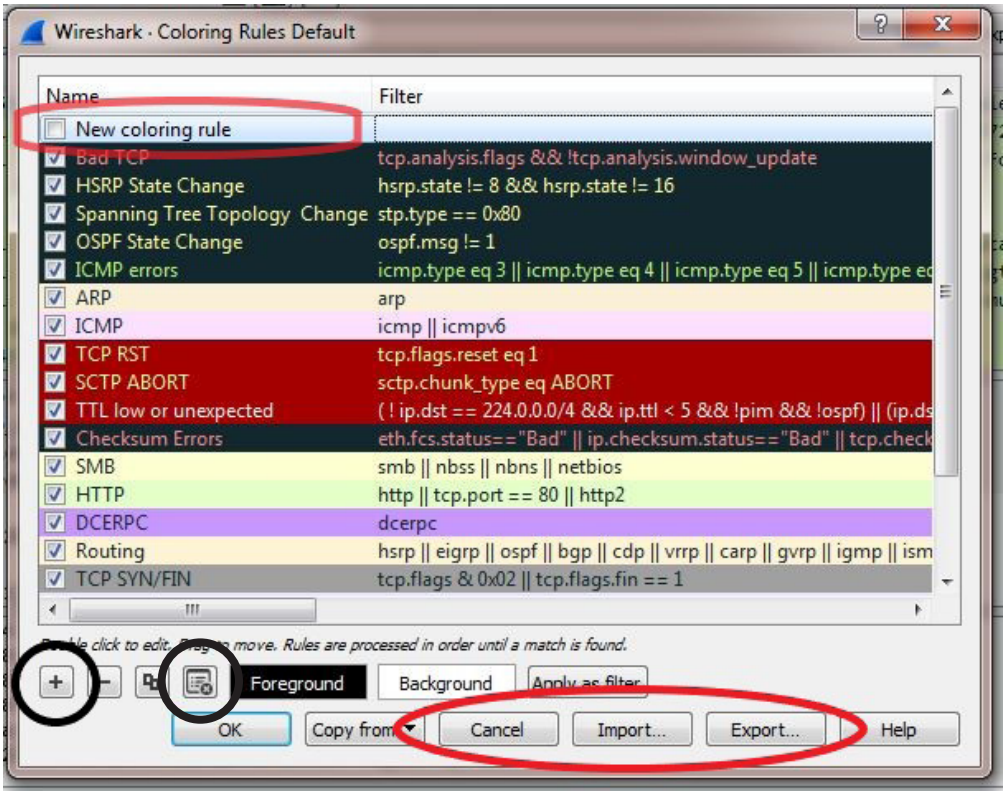
Custom color rules can be created and are useful in highlighting certain types of network traffic.

To create a new color rule, open the Coloring Rules window, and then select the plus icon. This will open the window to allow a new rule to be created. Choose a name and the display filter for the desired connection. Customize the foreground and background colors to highlight the traffic.



Wireshark's Edit Color Filter window.

Color sets can be created to match any number of specific needs and imported into the current set. When a set is imported, it is simply added to the current set and does not replace the current configurations.



Wireshark's Coloring Rules window.

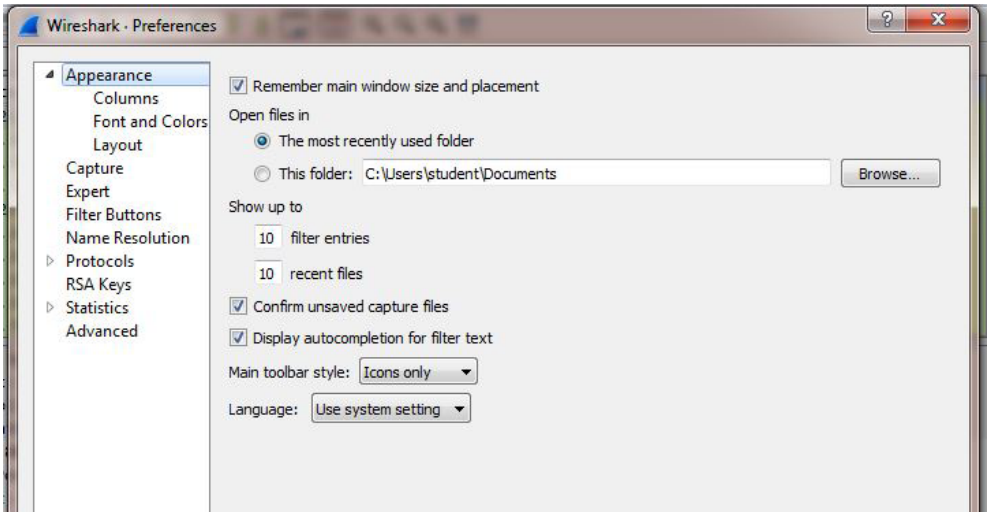
- Export takes all current rules and saves the setting out to a file.
- Import adds the desired configuration file to the current configuration.
- Clear removes all custom rules and changes the color settings back to the global defaults.

Wireshark Preferences

Preferences allow users to customize the software to best suit the needs of the situation. Depending on the specific tasks or job duties, setting up the system in different configurations may help.

Different preferences are available to users. These may help with using the application, analyzing data, or improving system performance.

To open Wireshark's Preferences window, navigate to Edit and then Preferences. The headings on the left allow users to customize various options such as the layout of the individual panes, column fields and their order, capture device and filters, and protocol preferences.

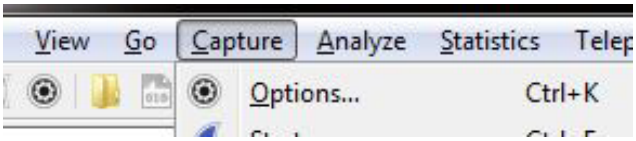
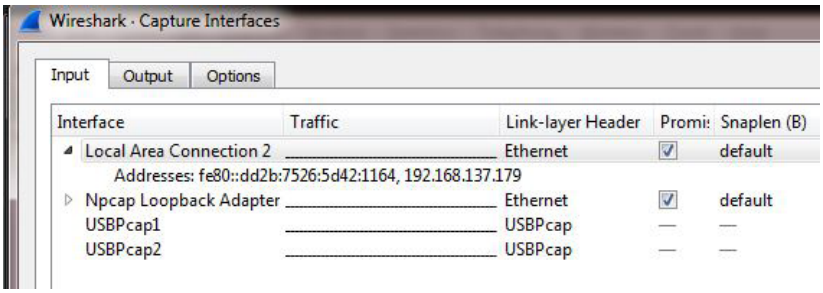


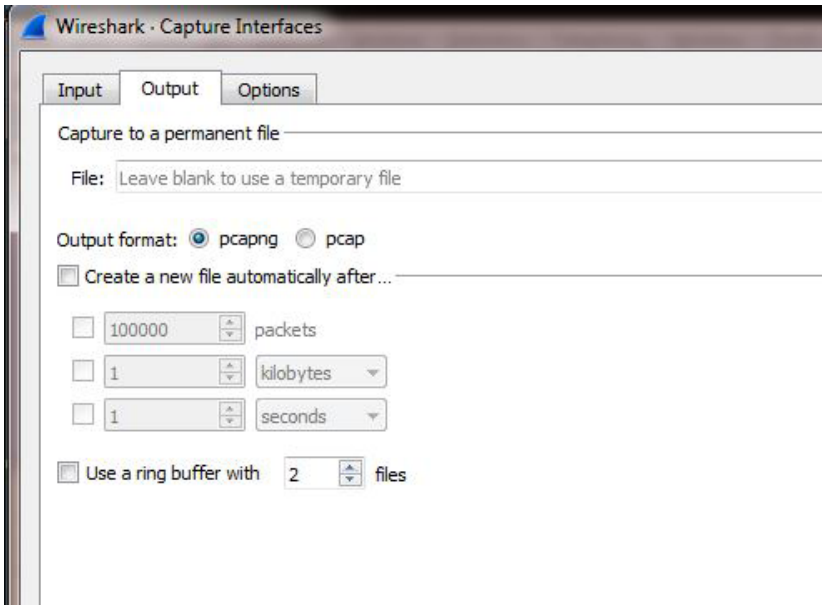
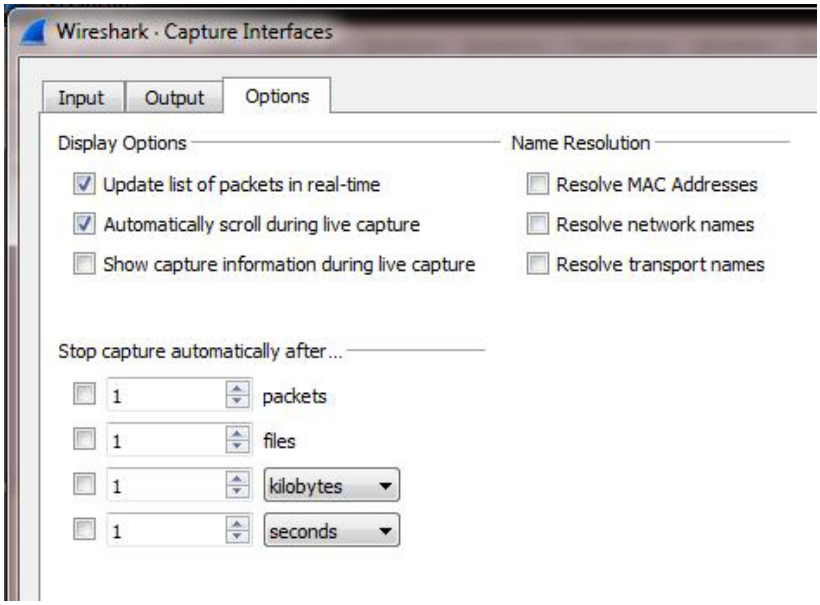
Capturing With Wireshark

Before beginning to capture data with Wireshark, certain configuration options are needed to ensure that the required data is being captured correctly. The next procedure highlights the important options that need to be set or change before capturing data.

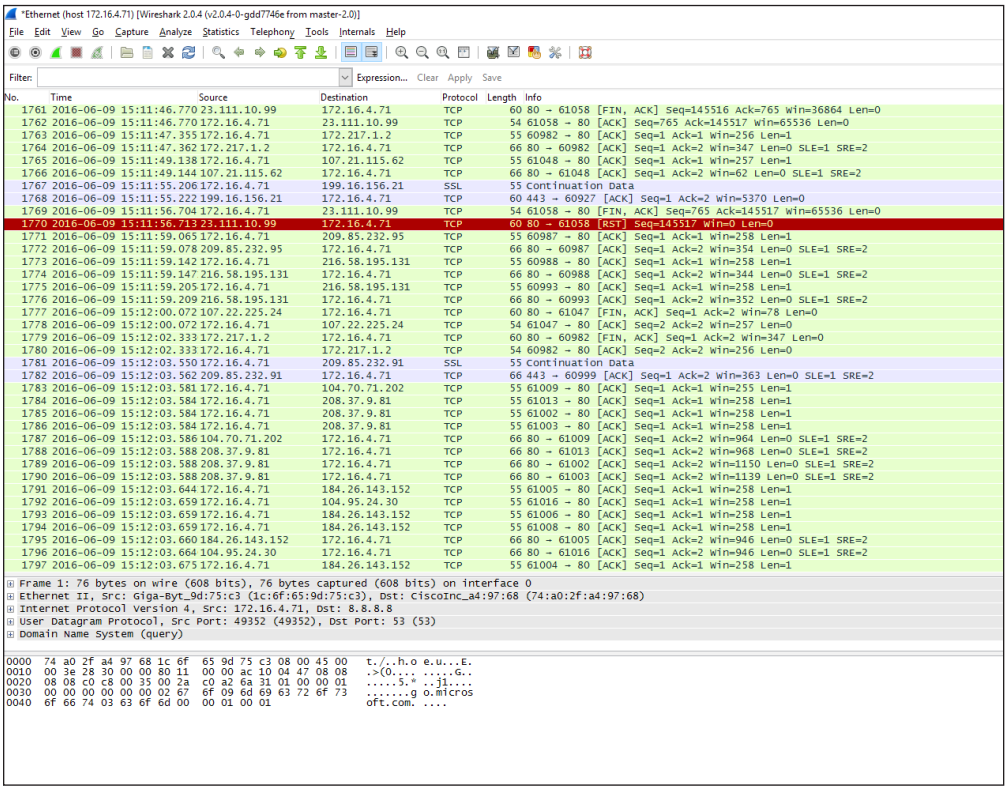
Procedure: Capturing With Wireshark

Follow these steps to begin capturing data with Wireshark.

Step	Action
1	Select Capture and then choose Options. 
2	The Input tab shows available NICs. Ensure that the Promiscuous box is checked next to the NIC that will be used for the capture. 
3	Click on the "Output" tab to modify how the pcap will be saved.
4	Click on the "Browse…" to determine where you will save the pcap.
5	Name your file "mycapture".
6	Select the either the pcapng or pcap radial option. The pcapng has more features than the pcap format. For this walkthrough, either option should work.

Step	Action
7	Both the "Create a new file automatically after…" and the "Use a ring buffer with" options should remain unchecked. The "Use a ring buffer with" option will cause Wireshark to write over previously saved data. 
8	Ensure that the boxes in the Options tab reflect the provided example. 
9	Click Start to begin the capture process.

Full-content Capture With Wireshark



Once the capture process begins, Wireshark displays a screen that is similar to the one here.

Wireshark is now performing a full-content capture. Information about the packets and the packets' content is being saved to the capture files. Notice that the packet information scrolls by rapidly and that packets are being received from various sources.

Filters

Wireshark offers several filtering and searching options:

- **Display filters:** Interface used to filter traffic currently being displayed by Wireshark
- **Capture filters:** Interface used to filter data while it is being captured from a network; uses the tcpdump (Berkeley Packet Filter - BPF) syntax

Display Filters

Display filters allow investigators to focus on the packets of interest while still saving those packets that do not match investigators' criteria. Display filters can be initiated while the capture process proceeds in the background.

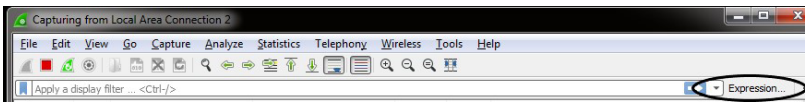
Display filters allow investigators to select packets by:

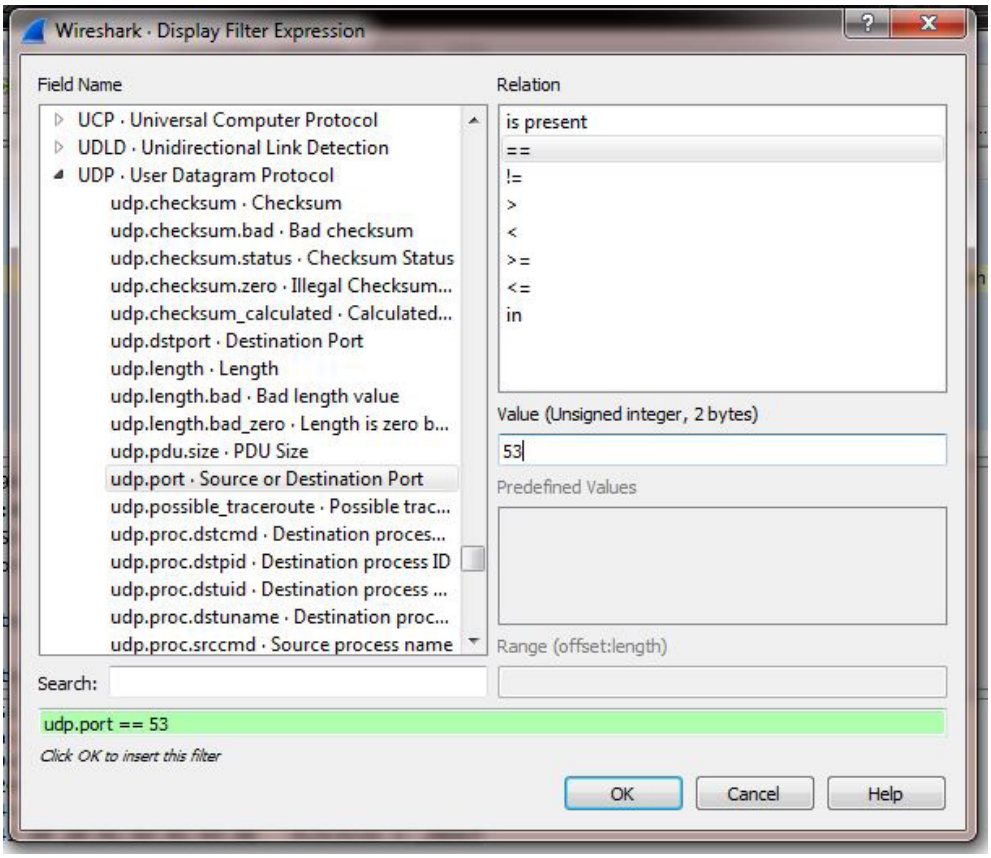
- Protocol
- Source and destination addresses
- MAC addresses
- Many other options

Procedure: Creating a Display Filter in Wireshark

Follow these steps to create and apply a display filter in Wireshark.

Note: This filter is used for data being displayed in Wireshark. Only the displayed data is changed, not the content of the log.

Step	Action
1	Click the Expression button, toward the upper right corner of the Wireshark window. 
2	When the "Wireshark: Filter Expression" window displays, scroll down in the "Field Name" pane until you see the protocol that is targeted for filtering.
3	Left click the arrow beside the protocol name one time to expand the menu of options.
4	Scroll down farther, locate the protocol option on which to place a filter and left click it. The panes may change to reflect options available for that protocol.
5	In the Relation pane of this window, select the desired option by left clicking it one time.
6	If needed, enter a value into the "Value" dialog box. Click OK Note: The example below creates a filter to only show UDP traffic with a source or destination port of 53.
7	You will be returned to the Capture. The filter box should be highlighted green with your newly built filter.
8	To the right of the filter line is an arrow that is highlighted blue. Click the arrow to apply the filter.



Directly Entering Display Filter Expressions

When a filter expression is created using the Display Filter wizard, the text for the filter is entered into the Display Filter field in the main window of Wireshark. Filter expressions can also be directly entered as text into that field instead of using the wizard.

If the display filter has a valid syntax, the background color of the display filter field will be green. Otherwise, it will be red.

Syntax of Display Filters

Wireshark display filters use a different syntax than tcpdump. The available protocols and filtering options are extensive and cannot all be listed in this text. Here are some common examples:

Operation	Syntax	Example
Source or destination IP address	ip.addr == <address>	ip.addr == 192.168.0.1 ip.addr == 20ab:5183:4383:::2ff:fee2:7596
Source IP	ip.src == <address>	ip.src == 192.168.0.1 ip.src == 20ab:5183:4383:::2ff:fee2:7596
Destination IP	ip.dst == <address>	ip.dst == 192.168.0.1 ip.dst == 20ab:5183:4383:::2ff:fee2:7596
Source or destination port number	tcp.port == <number> udp.port == <number>	tcp.port == 80 udp.port == 53
Source port	tcp.srcport == <number> udp.srcport == <number>	tcp.srcport == 80 udp.srcport == 53
Destination port	tcp.dstport == <number> udp.dstport == <number>	tcp.dstport == 80 udp.dstport == 53
Protocol	<protocol>	icmp

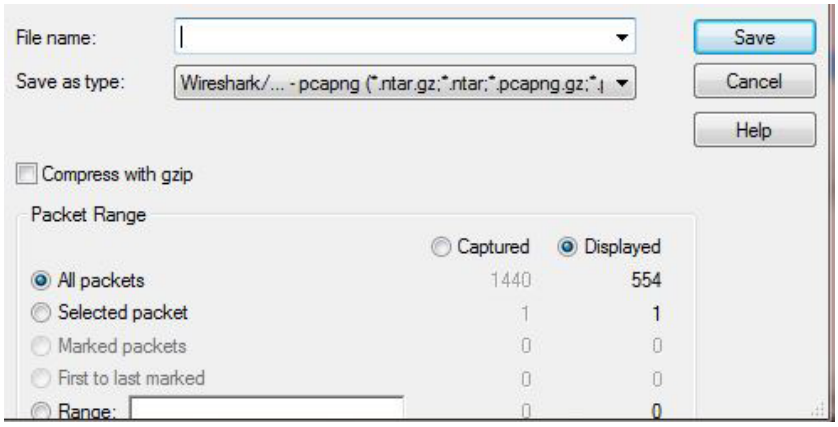
Altering and Combining Expressions

Expressions can be combined in the Display Filter Field, and logical operations can be performed on them. Allowed grouping and logic operators include:

Operation	Syntax	Example
Combine two expressions	<Filter 1> and <Filter 2>	ip.addr == 192.168.0.1 and tcp
Negate an expression	! <Filter 1>	! ip.addr == 10.0.0.4
Alternate expressions	<Filter 1> or <Filter 2>	ip.addr == 10.0.0.4
or ip.addr == 10.0.0.5		
Compare to value with "lesser than"	<Filter 1> < <Value>	tcp.port < 1024
Compare to value with "lesser than or equal to"	<Filter 1> <= <Value>	tcp.port <= 1024
Compare to value with "greater than"	<Filter 1> > <Value>	tcp.port > 1024
Compare to value with "greater than or equal to"	<Filter 1> >= <Value>	tcp.port >= 1024
Compare to value with "equal to"	<Filter 1> = <Value>	tcp.port == 1024
Group expressions with parentheses	(<Filter 1> <Operator> <Filter 2>)	tcp.port 80 or (icmp or arp)

Procedure: Saving Filtered Data

Follow these steps to save the filtered data in a capture file.

Step	Action
1	Apply a display filter of your choice.
2	Once the filter has been applied, click File and then Export Specified Packets.
3	An "Export Specified Packets" window will appear.
4	Enter an appropriate filename.
5	Before saving the file, make sure the "All packets" radio button is selected and the "Displayed" selection button is enabled. For greatest compatibility with other packet manipulation applications, be sure "File type" is set to "Wireshark/ tcpdump/... - libpcap". 

Capture Filters

By default, Wireshark captures all data packets provided by the NIC. This is usually not a problem on a small to mid-sized network. However, on a large to enterprise-sized network, the amount of data collected can quickly become overwhelming.

To resolve this problem, Wireshark is equipped with capture filters that allow investigators to capture only the types of packets of interest.

Wireshark capture filters uses the same BPF syntax as many other network sniffers. These filters offer many ways for a user to specify the type of traffic gathered as they are discovered. Some of the search criteria available include host name, host address, MAC address, protocols and Web addresses.

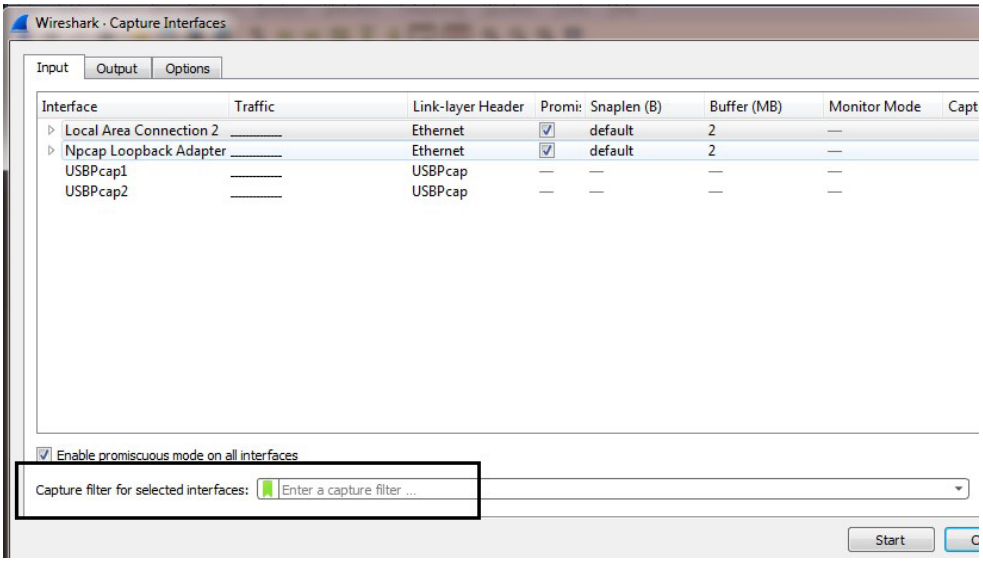
Procedure: Setting Up a Capture Filter

Follow these steps to see how Wireshark can filter data while it is being captured. This filter menu uses standard BPF syntax. Data filtered out through this method never gets stored.

Step	Action
1	Double click appropriate network interface in the Capture Options window to open the Edit Interface Settings dialog box.

2	Enter the following filter to capture all packets coming to and from a specific IP address and click OK to accept. Alternate capture filters can be substituted depending on your monitoring requirements.
	
	`host <IP Address>`
	
	Example:
	
	`host 192.168.31.144`
	
	This will capture all data coming from and going to IP address 192.168.31.144.

3	Click Start to begin capturing.

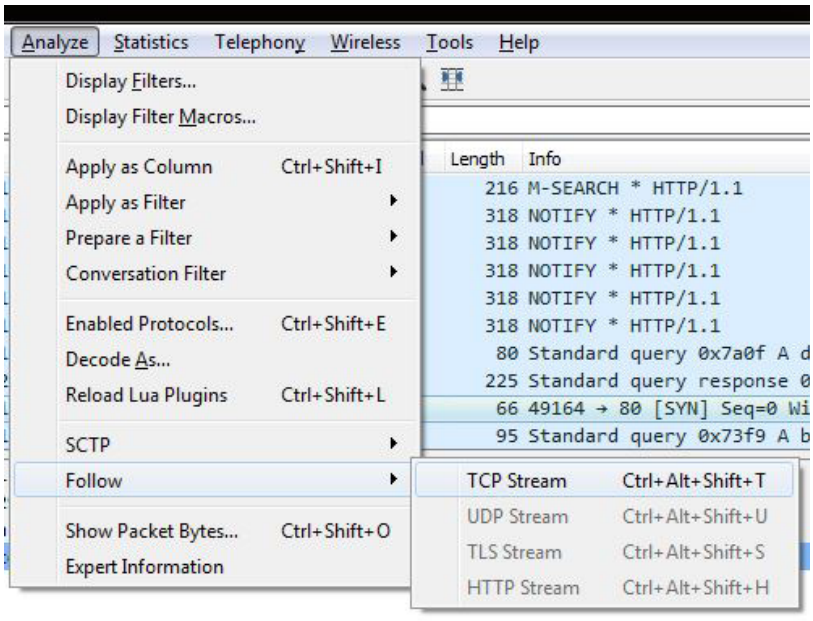


The TCP Stream

When working with TCP packets, viewing the packets of data the way the applications see them may be beneficial. Using the "Follow TCP Stream" will collect all packets associated with a session and display them in chronological order. This feature not only filters pertinent packets but also lists them in chronological order and displays the packet content.

Procedure: Viewing a TCP Stream

Follow these steps to use the "Follow TCP Stream" function.

Step	Action
1	With Wireshark running, find a captured packet that used the TCP protocol. Hint: Sort the packets by protocol by clicking the Protocol column heading. 
2	Right click the TCP packet and choose "Follow TCP stream."
3	A new window displays listing all associated packets in chronological order. Wireshark offers several viewing options in the "Follow TCP stream" window.

Lesson 3

Capturing Data With tcpdump

tcpdump is a universal command line sniffer that can be found on numerous OSs, including Linux, UNIX, Solaris and Mac OS X. The Windows compatible version of tcpdump is WinDump and offers all the same utilities of tcpdump. tcpdump has a steeper learning curve due to its strict command line execution. However, tcpdump offers a major benefit of using little system overhead. Additionally, its command-line interface affords the opportunity to include tcpdump into scripts as well as piped and redirected input and output. A network monitor can run tcpdump efficiently on less hardware power with more capturing options than other monitoring solutions.

This lesson introduces how to use tcpdump, including the most commonly used switches and options. As with any software, different versions of tcpdump exist, and each may have slightly different options and settings. Refer to the man page and become familiar with your version of tcpdump before using it in a production environment.

In this lesson, students learn the basic syntax of tcpdump and use it to capture network traffic.

OBJECTIVES *After completing this lesson, students will be able to:*
Describe the command line options of tcpdump
Conduct trap-and-trace network monitoring using tcpdump
Conduct full-packet capture using tcpdump

tcpdump Options

These options affect how tcpdump creates capture logs of network activity. The -w and -s switches are the most important, with the -C switch close behind. The -w switch tells tcpdump to store both the connection information and the actual data into a file. The -s switch tells tcpdump how much of the packet should be captured. The -C switch is used in conjunction

with -w to save captures as multiple, sequential capture files instead of one large file.

By default, -s specifies a packet snap length in tcpdump of 65,535 bytes. To capture a smaller portion of each network packet, this option must be set to the desired byte length. Setting -s to "0" also specifies a snap length to the default maximum value of 65,535 bytes.

Command Option	Description
-w	Write to a file
-s	Specify number of bytes of each packet to capture; "0" specifies full content; tcpdump will capture full content by default if this option is not specified
-r	Replay a capture file or read a capture file
-i <eth0>	Monitor network data on the specified interface
-C <nnn>	Create a new, sequentially numbered capture file every <nnn> megabytes

Example:

```
tcpdump -i eth0 -s 0 –w capture.pcap –C 1900
```

Display Options

tcpdump has display options that only affect the way the connection information is displayed to the screen. These options do not affect how the data is stored in the file. The only exception to this is a trap-and-trace capture, which is explained later. A full capture can be replayed with different display switches in order for an examiner to see the data in a new way.

Command Option	Description
-A	Print the content of each packet in ASCII, except for the Data Link layer
-n	Do not convert numbers to names, such as port numbers to service names, or IP addresses to host names
-tttt	Print the date as the first field of the packet before the time
-v, -vv, -vvv	Print more verbose output, progressively increasing with more v's
-X	Print the data portion of each packet in hex and ASCII
-e	Print the data link header addresses (MAC addresses)

Example:

```
tcpdump -r capture.pcap -tttt -n -X
```

Trap and Trace Using tcpdump

tcpdump provides the capability of only logging the connection information of network communications. This is useful for trap-and-trace operations where the content of the network data is not needed. If using this capture technique, the content of the packets will never be captured and this information cannot be retrieved at a later date.

Do not use the -w option to write the information to a file because this can potentially capture data. Instead, use the redirect command to send the textual data to a file. Make sure the desired formatting options (how it is displayed to the screen) have been picked because these cannot be adjusted later as a binary capture can.

Example:

```
tcpdump –i eth1 –n –tttt –e > capture.txt
```

Statistical information (packets captured, packets filtered and packets dropped) can be added by appending tcpdump's error status to the text file used to log connection information:

```
tcpdump –i eth1 –n –tttt –e > capture.txt 2>> capture.
txt
```

This technique still allows tcpdump to "touch" full packet data even though only a text summary of network traffic is being saved. Depending on the scope of investigation, the inclusion of a "-s" tcpdump option may be included to ensure the data portion of network traffic is not a consideration.

```
tcpdump -i eth1 -n -tttt -e -s 96 > capture.txt 2>>
capture.txt
```

tcpdump Example

Here is an example of tcpdump output, broken out with line numbers and field descriptions.

Command: `tcpdump -i eth0 -tttt -e -n`

```
2013-07-19 20:28:10.287575 00:0c:29:bd:83:c1 >
00:50:56:fc:f7:1d, ethertype IPv4 (0x0800), length 54:
192.168.73.131.57407 > 74.125.228.106.80: Flags [.], ack
42781, win 36500, length 0
```

Line	Data	Description
1	`2013-07-19 20:28:10.287575`	Full date and time for that packet
2	`00:0c:29:bd:83:c1 >00:50:56:fc:f7:1d`	Source MAC address (00:0c:29:bd:83:c1) Destination MAC address (00:50:56:fc:f7:1d)
3	`192.168.73.131.57407 >74.125.228.106.80`	Source IP address (192.168.73.131) Source port (57407) Destination IP address (74.125.228.106) Destination port (80)
4	`ack 42781, win 36500, length 0`	TCP flags and settings

Full Packet Capture Using tcpdump

Capturing full network traffic via tcpdump is as simple as either omitting the "-s" option altogether or specifying "-s" as 0 and including the "-w" option to write the network traffic to a file. When using tcpdump to conduct full-content network captures, less attention needs to be given to display filters since the full content can be displayed in a multitude of ways during analysis.

tcpdump full capture example command:

```
tcpdump -C 1900 -i eth0 -n -s 0 -w /home/student/
evidence/capture.pcap host 10.1.1.12 and port 80
```

tcpdump Filter Syntax

tcpdump provides a set of filters that can be used during a network capture, or to analyze a saved capture file. Due to tcpdump's popularity within the security community, other tools have adopted the same filtering syntax for their network monitoring or log display functions. Applications that use the tcpdump filter syntax include:

- Wireshark
- ngrep
- jpcap
- Snort

Examples:

```
tcpdump –i eth1 –s 0 –w capture.cap –n host 10.1.1.2

tcpdump –i eth1 –s 0 –w capture.cap –n host 10.1.1.2 and
port 80

tcpdump –i eth1 –s 0 –w cap2.cap –e ether host
00:11:22:33:44:55
```

Many filter arguments can be specified and implemented simultaneously within tcpdump, but the tcpdump command can grow to an unmanageably large string quickly. This behavior can be changed with the -F option to specify a separate flat text file that contains filter expressions. Specifying a -F option would turn the following example from:

```
tcpdump -C 1900 -i eth1 -n -s 0 -w /home/student/
evidence/capture.pcap host 10.1.1.12 and \(port 80 or
port 53 or port 22\) and not net 74.125.228.0/32
```

into:

```
tcpdump -C 1900 -i eth1 -n -s 0 -w /home/student/
evidence/capture.pcap -F/home/student/filter.txt
```

where the file "filter.txt" in the directory /home/student/ contained:

```
host 10.1.1.12

&& port 53

|| port 22
```

```
|| port 443

&& !net 74.125.228.0/32
```

This type of filter implementation can be useful when calling tcpdump with a generic command within a script with reference to custom filtering. The filtering can easily be changed without having to edit the executable script.

Lesson 4

Retrieving Captured Data

Once data has been successfully collected by a network monitor, it must be analyzed for content. This allows investigators to tie specific actions to the subject of the investigation. Most often, particularly with remote installations of a network monitoring station, captured data must be retrieved to an analysis workstation.

This lesson demonstrates different methods of moving the captured data from the monitoring machine for analysis.

OBJECTIVES *After completing this lesson, students will be able to:*
Copy captured data over the network via a shared directory
Use the FTP service to copy captured data from the monitoring system
Use the SCP service to copy captured data from the monitoring system

Data Retrieval

Once investigators have stealthily gathered data, they do not want to lose that advantage by revealing their presence with a large amount of unrecognized traffic (either physical or electronic). Their goal is to leave without disruption and only with the captured data.

Reasons to Retrieve (Exfiltrate) Data

Many reasons exist to move the data from the network monitoring workstation. It would be easiest to move the data and the workstation when finished with the capture, but investigators may want to move partial data during capture. Among the options:

- Move/duplicate the data for safekeeping purposes away from the network monitoring workstation
- Move the data for analysis at another site
- Move the data to send progress/status reports

Ways to Retrieve Capture Data

Two basic methods exist to retrieve captured data: removing it on a physical medium or sending it electronically. The most common methods are:

- External media such as a USB device
- External optical media such as a CD or DVD
- Data transmission over the network
- Data transmission over a crossover cable

Avoiding Detection

When transmitting data over the network, investigators must ensure their presence on the network is not detected. The best way to accomplish this is by using an out-of-band communication. Using a crossover cable without a second NIC requires that the capture machine be physically unplugged and plugged back into the network. Detection might be avoided, but the ability to continue to capture data will be lost.

Moving the data to physical medium is another method to avoid network detection. This methodology, however, has drawbacks because an unknown person carrying external hard drives or optical media is often considered suspicious behavior.

Using a Shared Directory

A shared directory on the management computer is a simple way to get files transferred in real time or copied over at a later time. When saving the capture files directly to the shared directory, make sure the bandwidth can handle the data so that information is not lost.

The shared directory needs to be on the management machine so that the monitor station can access it.

SMB/CIFS

Even though Server Message Block (SMB) became Common Internet File System (CIFS) years ago, the old acronym has been maintained by the IT community. This protocol will sometimes be called either name or both.

SMB/CIFS is the protocol created by Microsoft for file sharing between Windows computers. While not typically used between computers on the Internet, it is the most common file-sharing protocol on private networks.

This lesson focuses on setting up a Samba share so that it is accessible by both Windows and Linux machines. Samba is the application used to set up the share. SMB is the protocol used on the wire to access a Samba share.

File-transferring Protocols: FTP and SCP

An FTP server can also be set up to receive data from remote monitors. FTP has the advantage over a file share of being able to transfer data across multiple networks (network shares are usually confined to a LAN). However, FTP does not encrypt any of the data, including credentials, across the wire. Using an encryption method like SCP to send the data is recommended. To connect to an FTP server, use the same procedure as Samba, but change Windows share to Public FTP in the Service Type menu. SFTP or Secure File Transfer Protocol is a secure option for file transfer that uses common FTP commands over an SSH connection.

Secure Copy (SCP) uses SSH functionality to send encrypted files over a network. Instead of getting a shell on the remote machine, it is designed to transmit data. It is one of the most secure ways to transfer data because the entire session is encrypted. To use SCP, the server must have SSH server running and be accessible through the firewall. This can be done by running SSH under the Services menu and enabling SSH on the firewall.

MODULE 5

Analysis

The overall goals of this course focus on monitoring and capturing network traffic, but keeping a bigger picture in mind is also important. Why are we collecting a certain set of network traffic in the first place, and how can we quickly and efficiently determine if the data we are collecting fits our mission goals, scope and intent? What is the context of the network traffic we are monitoring and capturing? Can we better hone our filters to more efficiently capture what we want, not capture what we don't want and not miss something important?

This module illustrates the characteristics and significance of different types of data captured over a live network. In many instances, live data transmitted over a computer network can provide the missing link between data recovered through traditional dead-box forensics and positive linkage to an actual act.

Once network traffic is captured, a brief analysis of the capture file should be conducted. This analysis may reveal the presence of additional targets or indications of a network intrusion that may affect an investigation. Based on that information, additional captures may be required. Students learn the basics of analyzing network traffic using Wireshark, NetWitness Investigator and Snort. The basic functions of these programs are introduced by analyzing capture files containing Web and file transfer protocols.

OBJECTIVES *After completing this module, students will be able to:*
Analyze network traffic
Analyze web traffic
Analyze file transfer traffic
Explain how tcpreplay is used to analyze network traffic
Explain how tcpflow is used to analyze tcp conversations
Analyze network traffic using an intrusion-detection system (IDS)
Analyze network traffic and system artifacts to identify probing and intrusion techniques
Recognize Attacker Command & Control (C2) techniques

Lesson 1

Traffic Analysis

Traffic analysis can examine packets from a high-level view or in detail. Areas of interest at a high level are the types of traffic in the capture, the endpoints in the capture and established sessions.

This lesson introduces students to the basics of captured packet analysis using Wireshark and NetWitness Investigator.

OBJECTIVES *After completing this lesson, students will be able to:*
Use Wireshark and NetWitness Investigator to examine captured network traffic
Configure and employ filters in Wireshark and NetWitness Investigator
Conduct targeted searches in Wireshark and NetWitness Investigator

Wireshark

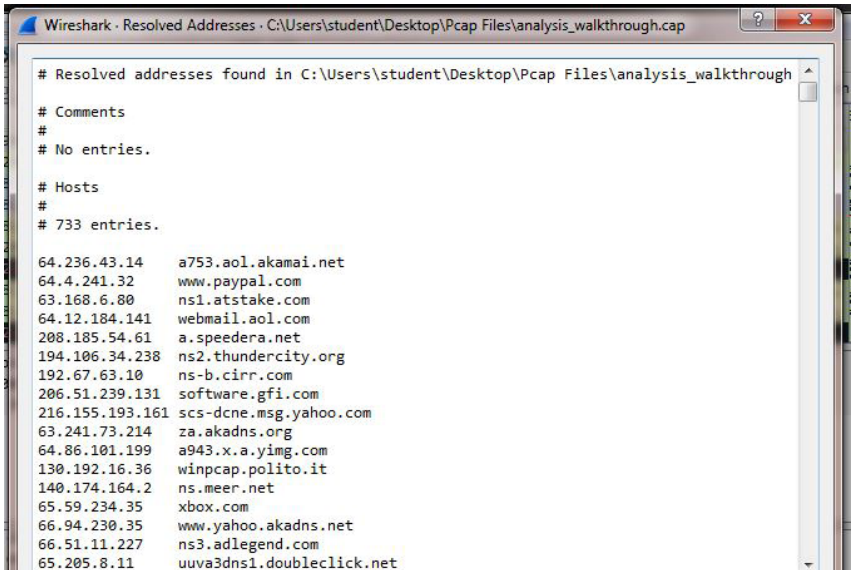
The protocol analyzer Wireshark can be used to analyze network traffic capture files. Wireshark performs live network capture and offline analysis using a three-pane packet browser interface. Support is provided for hundreds of protocols, including SSl/TLS and VoIP.

Wireshark has capabilities that allow it to display high-level information about the content of a capture file. These functions fall under the Statistics menu. The Show address resolution view displays all resolved domain names associated with IP addresses in the current capture file.

The Statistics Endpoints view shows each endpoint address, along with number of packets and bytes transmitted and received. This view easily identifies every connection from a suspect computer. The endpoints display segregates the packets, which allows the user to display just the Ethernet, TCP, UDP, IPv4 and IPv6 packets. The displayed data can be sorted by clicking on any of the columns.

Procedure: IP Address Resolution

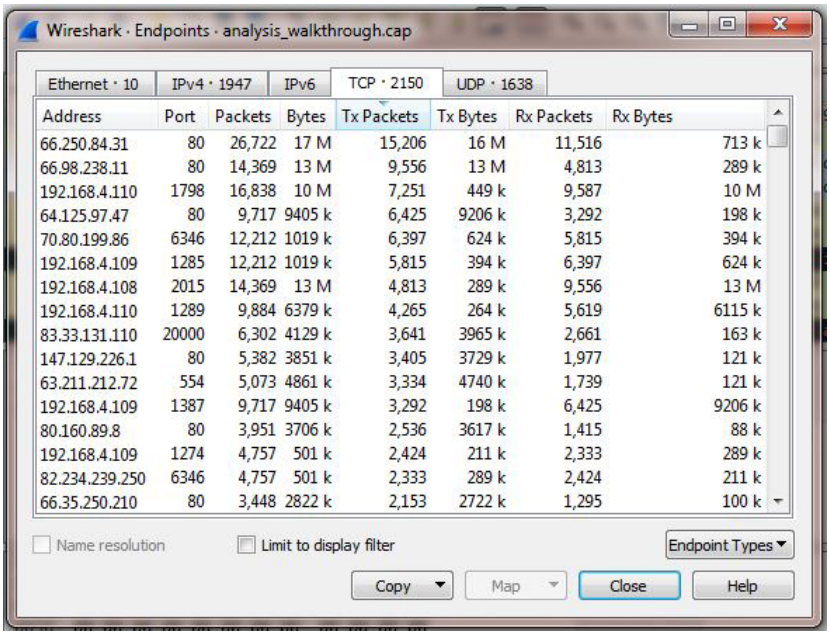
Follow these steps to list the IP addresses to domain name resolution for the current capture file.

Step	Action
1	Open the Wireshark application.
2	The main window will have files listed under the Open section. Select a file as directed by the instructor. The file may take some time to load.
3	Click Statistics on the menu bar. Click Resolved Addresses. 

Procedure: Generating Protocol Endpoint Display

Follow these steps to generate and display a listing of all the endpoints found in the currently open capture file.

Note: Endpoints are the logical source or destination of a network communication. The display is categorized by network protocol.

Step	Action
1	Open the Wireshark application.
2	The main window will have files listed under the Open section. Select a file as directed by the instructor. The file may take some time to load.
3	Click Statistics and then Endpoints. The capture file will be scanned, and a list of all endpoints will be generated, along with statistics about each endpoint. Select TCP from the row of tabs at the top of the endpoints display. Then click at the top of the Packets column to sort the list by size. 

Wireshark Network Conversations

Network conversations can be one-way or bidirectional. The Conversations view, listed under the Statistics menu, shows every conversation in the capture file. Both endpoints are shown, along with protocols (ports), the number of packets, and the throughput of data.

Categories are available as in the Endpoint display. This display shows the nature of the conversation and its extent in time and data exchanged and is sortable on any column.

Procedure: Displaying Network Conversations

Follow these steps to display network conversations.

Step	Action
1	Use the file already open in Wireshark.
2	Click Statistics on the menu bar. Then click Conversations.
3	The capture file will be scanned, and a list of all endpoints will be generated, along with statistics about each endpoint. Select TCP from the row of tabs at the top of the endpoints display. Then click at the top of the Packets column to sort the list by size.

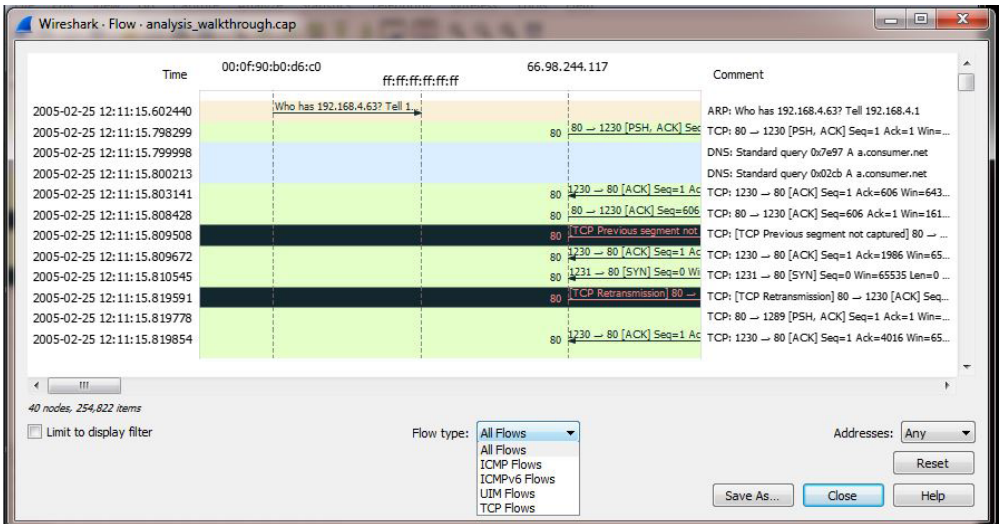
Flow Graph

The Flow Graph is available under the Statistics menu. The Flow Graph depicts network traffic in a table view and creates a display of captured file traffic by distributing addresses across columns and packet detail across the rows. The packets are chronological in the vertical direction. The Flow Graph function can be used at a high level for an overview of traffic, but it also contains enough detail to facilitate deeper analysis of the data, including time studies. Additionally, as different rows are selected, the corresponding packet is highlighted and displayed in the main Wireshark window.

Procedure: Generating a Flow Graph Display

Follow these steps to generate a Flow Graph display.

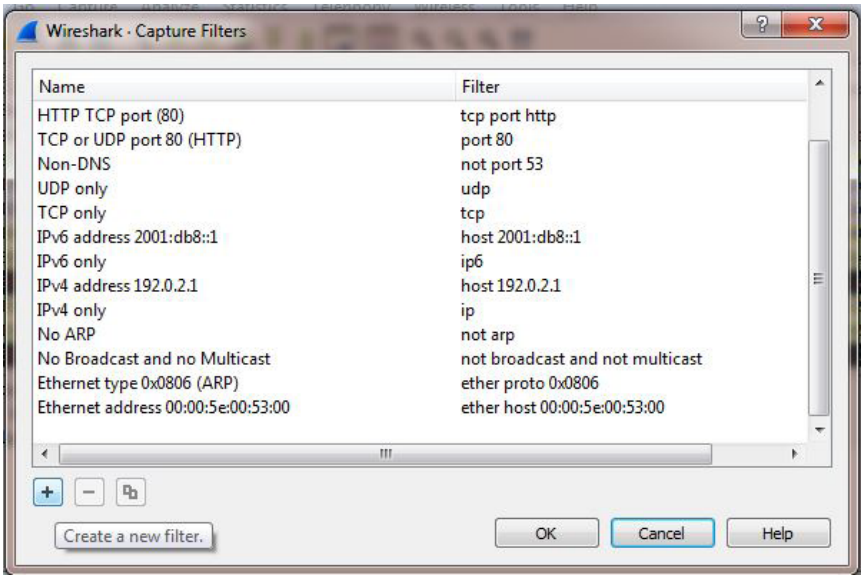
Step	Action
1	Use the file already open in Wireshark.
2	Click Statistics on the menu bar. Click Flow Graph. Select an option from the drop-down menu Flow type to restrict the graph as desired. Select All Flows unless only TCP Flows graphing is needed.
3	The capture file will be scanned, and the graph will be generated and displayed.
4	This graph may be saved with the Save As… option.



Procedure: Filtering for an IP Address

Follow these steps to filter for an IP address.

Step	Action
1	From within the Capture menu, select capture filters.
2	Click the plus icon to add an IP address to filter the capture.



Decoding the Display

In jpcap, each red square represents a host IP address. The red squares can be moved around the screen, allowing for additional organization. Other color codes include:

- Yellow traffic represents TCP traffic.
- Green traffic represents UDP traffic.
- Blue traffic represents ICMP traffic.

Filtering Data in Wireshark

When used to capture network data, Wireshark can accept or reject packets based on a capture filter. The capture filter limits packet capture data to authorized traffic based on a search authority. A capture filter can be used to limit the size of the capture by rejecting uninteresting data.

When displaying network capture data, Wireshark also uses a display filter. The display filter changes the packets that are visible to the user without modifying the data in the capture file. The user may want to only see traffic belonging to a particular protocol, or may be interested in traffic associated with a given IP address during an analysis. The capture filter allows the visible data set to be limited as desired while maintaining the capture file intact. Wireshark uses a different syntax for display filters and contains a display filter expression builder.

Display Filtering in Wireshark

A display filter limits the scope of the displayed data set in Wireshark. The display filter can be created by typing it into the field or using an expression builder. The expression builder facilitates identifying the appropriate filter variables and operator syntax. The next table shows some of the common display filter variables.

Operation	Syntax	Examples
Source or destination IP address	ip.addr == <address>	ip.addr == 192.168.0.1 ip.addr == 20ab:5183:4383:::2ff:fe2:7596
Source IP	ip.src == <address>	ip.src == 192.168.0.1 ip.src == 20ab:5183:4383:::2ff:fee2:7596
Destination IP	ip.dst == <address>	ip.dst == 192.168.0.1 ip.dst == 20ab:5183:4383:::2ff:fee2:7596
Source or destination IP network	Ip.addr == <network address>	ip.addr == 10.0.0.0/8 ip.addr == 192.168.1.0/24
Source or destination port number	tcp.port == <number> udp. port == <number>	tcp.port == 80 udp.port == 53
Source port	tcp.srcport == <number> udp.srcport == <number>	tcp.srcport == 80 udp.srcport == 53
Destination port	tcp.dstport == <number> udp.dstport == <number>	tcp.dstport == 80 udp.dstport == 53
Protocol	<protocol>	icmp

Expressions can be combined in the Display Filter Field, and logical operations can be performed on them. Allowed groups and logical operators include:

Operation	Syntax	Examples
Combine two expressions	<Filter 1> and <Filter 2>	ip.addr == 192.168.0.1 and tcp
Negate an expression	! <Filter 1>	! ip.addr == 10.0.0.4
Alternate expressions	<Filter 1> or <Filter 2>	ip.addr == 10.0.0.4 or ip.addr == 10.0.0.5
Compare to value with "lesser than"	<Filter 1> < <Value>	tcp.port < 1024
Compare to value with "lesser than or equal to"	<Filter 1> <= <Value>	tcp.port <= 1024
Compare to value with "greater than"	<Filter 1> > <Value>	tcp.port > 1024
Compare to value with "greater than or equal to"	<Filter 1> >= <Value>	tcp.port >= 1024
Compare value with "equal to"	<Filter 1> = <Value>	tcp.port == 1024
Group expressions with parentheses	(<Filter 1> <Operator> <Filter 2>)	tcp.port 80 or (icmp or arp

Creating a Display Filter for a Keyword

A display filter can be created for a keyword. Unlike a keyword search shown later in this lesson, a keyword filter will change the display so that it only shows packets that contain the search term. This is done with the "frame contains" display filter, which can be used to filter for the presence of a keyword anywhere in a packet.

Creating a Display Filter for a Hex Value

The "frame contains" expression syntax can also be used to filter for hexadecimal values. For example, this expression could be used to display packets containing the hex value 0x6d73646f.

```
frame contains 6d:73:64:6f
```

The hex value is entered in place of a keyword, with colons used to separate the value into pairs.

Directly Entering Display Filter Expressions

When a filter expression is created using the Display Filter wizard, the text for the filter is entered into the Display Filter field in the main window of Wireshark. Filter expressions can also be directly entered as text into that field instead of using the wizard.

Any filter expression can be directly typed into the Display Filter field, for example "frame contains password" or "frame contains 6d:73:64:6f". Once users are familiar with the display filter syntax, this is the fastest way to generate and apply a display filter.

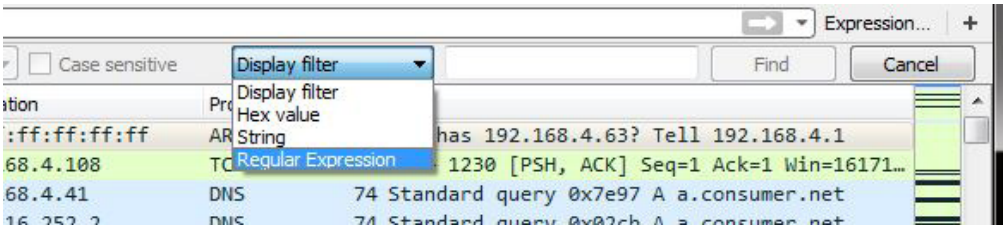
Searching in Wireshark

A standard search for text or hex data can be conducted in Wireshark. This function will not remove frames from the display in Wireshark like display filters do. Instead, the search will scan through the frames and highlight the first frame that matches the search criteria.

Procedure: Conducting a Search in Wireshark

Follow these steps to conduct a search in Wireshark

Step	Action
1	Select Edit in the menu bar. Select Find Packet from the drop-down menu.
2	A new menu bar will appear with a Display filter drop-down menu. Select the data type for the search from the drop-down options. Options include: • **Display filter:** Enter a standard display filter • **Hex value:** Enter a hex value as the search target • **String value:** Enter a string value as the search target
3	Enter a target value in the field next to the Filter button.
4	Click the Find button and the display will change back to the main Wireshark window. The first matching frame will be highlighted.



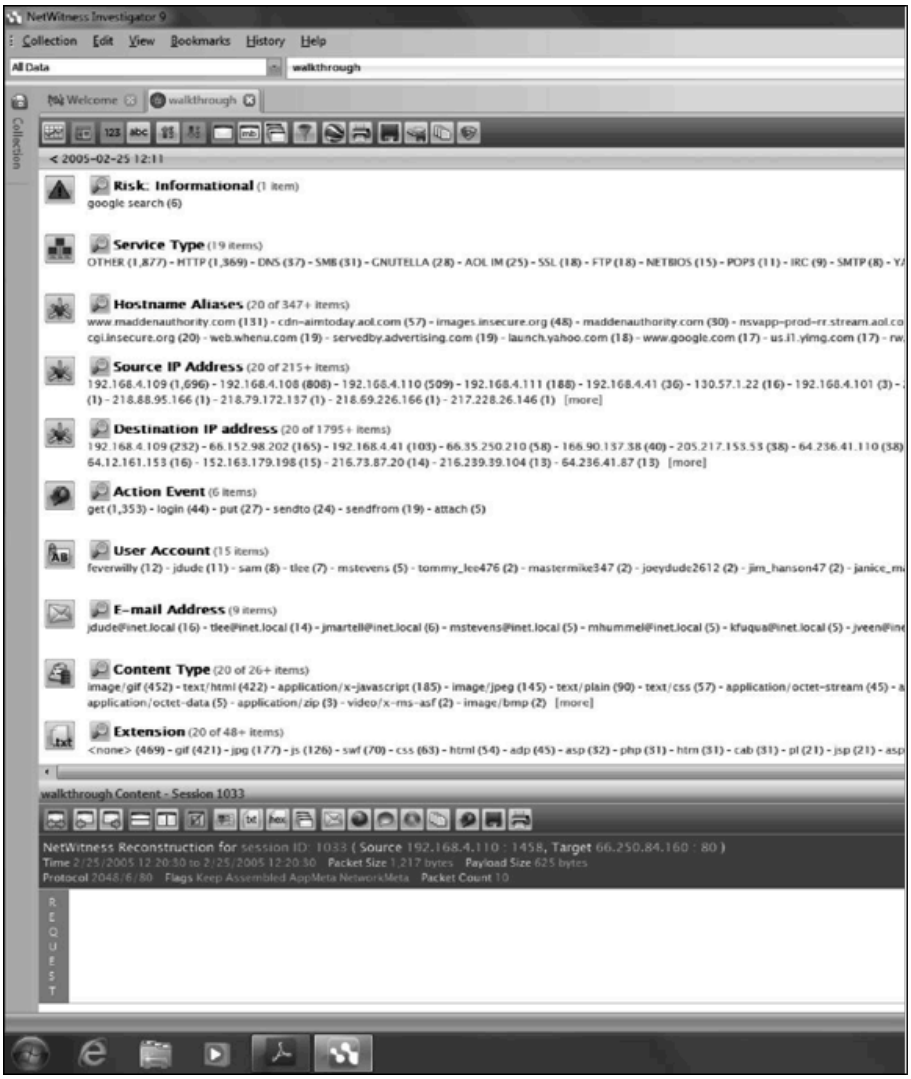
NetWitness Investigator

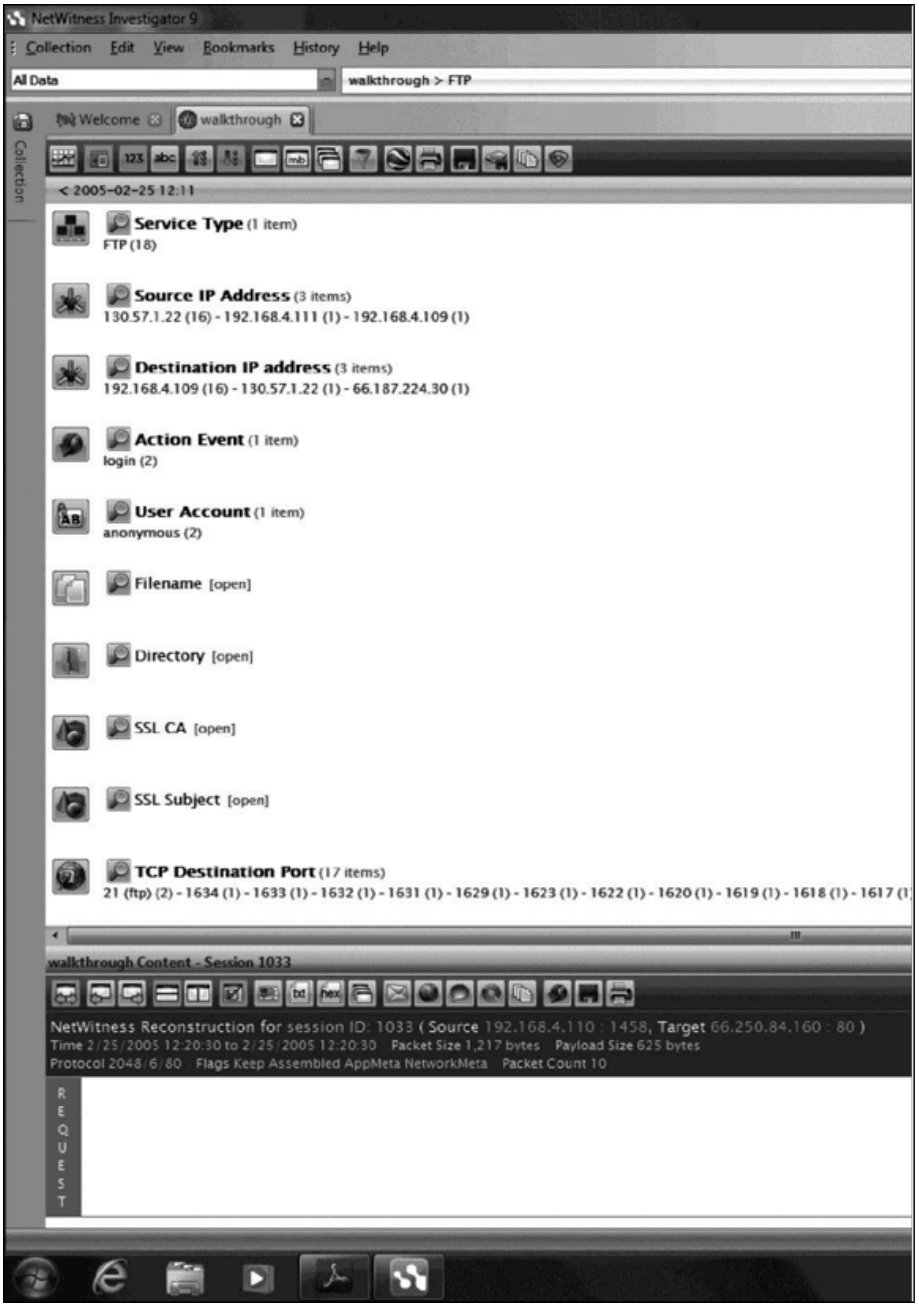
NetWitness Investigator provides free-form contextual analysis of live or recorded network capture data. The Investigator interface displays network data in a framework of application layer categories, including geographical location, protocols, addresses, files and user information.

Using Investigator

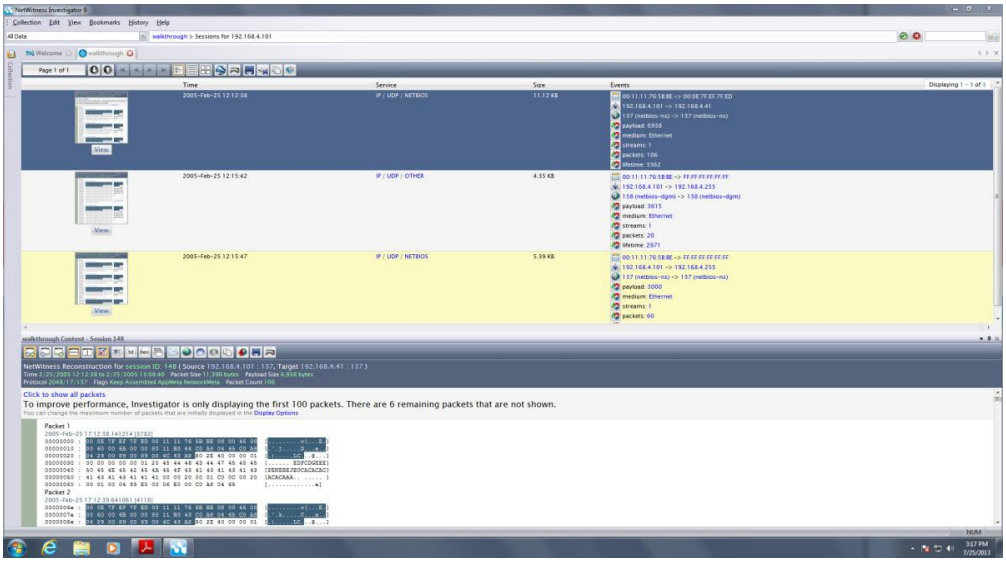
NetWitness Investigator has a different way of displaying data in a capture file. The main page for a capture file consists of high-level categories. These categories may have subcategories, containing capture file-specific entities that the user can drill down into.

The main page for a capture file in NetWitness Investigator.





In NetWitness Investigator, clicking a subcategory will open another top-level display showing filtered results for the selected subcategory. Clicking the number of occurrences of the subcategory will provide the capability to review each occurrence.



NetWitness Investigator panes. The session list pane (top left) shows each session. The session contents pane (top right) shows an overview of each session's content. The packet contents pane shows the content of the packets in the selected session.

Students will use Investigator to open the same capture file they looked at in Wireshark.

Procedure: Navigating through a Network Capture With NetWitness Investigator

Follow these steps to navigate through a network capture with NetWitness Investigator.

Step	Action
1	Click the NetWitness Investigator icon on the desktop.
2	Click Collection on the menu bar. Then click New Local Collection. Enter a name in the Collection Name field. Leave the path below the Collection Name as Netwitness\Investigations under the user Documents directory. Click OK. Notice that the new collection appears in the left column with the name just entered. To the right of the name there will be a -. Double click the collection name. The - should change to Ready.

Step	Action
3	Right click the collection name, and select Import Packets. An Open dialog box will appear. Follow the instructions given to locate the desired pcap file to import. During the import process, a progress bar appears in the left column.
4	Double click the collection name to open a tab to review that collection.
5	Note the upper-level categories Service Type, Source IP Address, Destination IP Address, all the way down to cities and Ethernet source and destinations. Each of these categories contains lists generated from the content of the capture file.
6	The main screen has the upper-level overview of the capture file content.
7	Click Collection on the Menu bar. Then click Summarize Collection. The display is a timeline showing session counts and session bytes across the timeline covered by the capture file.

tshark

tshark is the Windows Command prompt version of Wireshark. tShark is able to detect, read and write the same capture files that are supported by Wireshark. But since it is a CLI it does not require a Graphical User Interface (GUI) and is scriptable. It lets you capture packet data from a live network, or read packets from a previously saved capture file, either printing a decoded form of those packets to the standard output or writing the packets to a file. tShark's native capture file format is pcap format, which is also the format used by tcpdump and various other tools.

Without any options set, tShark will work much like tcpdump. It will use the pcap library to capture traffic from the first available network interface and displays a summary line on stdout for each received packet.

Useful flags	
-f <FILTER>	Specify a capture filter
-I <INTERFACE>	Specify interface/pipe to use for live captures
-t <FORMAT>	Specify format of timestamp ("ad" = absolute with date, "r" = relative)
-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: filed in Wireshark); packets that do not match are discarded
-S	Decode and display packets
-w <OUTFILE>	Write capture date to <OUTFILE>

tshark Example

Capture packets and display only packets from the "10.10.200.0/24"

```
# tshark -i eth2 –t ad -n -S -R "ip.
addr==10.10.200.0/24"
```

Read in file "webcapture.pcap" and display only HTTP traffic the string ".pdf" and save the results to a new file called "newpdfs.pcap"

```
# tshark -r webcapture.pcap -S -R "http and frame
contains `.pdf' " –w pdfs.pcap
```

Replaying Network Traffic:

Some analysis tools can accept a packet capture file (.pcap) as an input for offline analysis. However sometime the only way to understand how a system performs in response to real network traffic is to replay the packets.

For more details, please see the tcpreplay Manual at: http://tcpreplay.synfin.net/trac/wiki/manual

tcpreplay is a suite of utilities for UNIX and Win32 operating systems for editing and replaying network traffic which was previously captured by tools like tcpdump and Wireshark. The goal of tcpreplay is to provide the means for reliable and repeatable traffic for testing a variety of network devices such as switches, router, firewalls, network intrusion detection and prevention systems (IDS and IPS) . It is important to note that tcpreplay is completely stateless and is unable to handle updating TCP sequence and acknowledgement numbers, so it does not support replaying traffic to a server. However tcpreplay provides the tools to classify traffic as client or server edit packets at layers 2-4 of the OSI model and replay the traffic at arbitrary speeds onto a network for sniffing or through a device.

The tcpreplay suite includes the following tools:

- **tcpprep** - multi-pass pcap file pre-processor which determines packets as client or server and creates cache files used by tcpreplay and tcprewrite
- **tcprewrite** - pcap file editor which rewrites TCP/IP and Layer 2 packet headers

- **tcpreplay** - replays pcap files at arbitrary speeds onto the network
- **tcpliveplay** - Replays network traffic stored in a pcap file on live networks using new TCP connections
- **tcpreplay-edit** - replays & edits pcap files at arbitrary speeds onto the network
- **tcpbridge** - bridge two network segments with the power of tcprewrite
- **tcpcapinfo** - raw pcap file decoder and debugger

Example: Replay a given pcap as it was captured:

```
# tcpreplay   -i  eth0 sample.pcap
```

Example: Replay traffic as quickly as possible (Exercise)

```
# tcpreplay –topspeed -i   eth0  sample.pcap
```

Like Wireshark's ability to rebuild data streams there are command line tools available to reconstruct a network conversations between two specific endpoints.

tcpflow is a command line interface (CLI) that can parse, reassemble, and extract the payloads of any TCP stream it finds in a libpcap packet capture. Tcpflow is similar to 'tcpdump', in that both process packets from the wire or from a stored file. It's also similar to WireShark, in that both allow analysis of network traffic. But unlike either tcpdump or WireShark, tcpflow reconstructs thousands (or millions) of TCP connections at a time and saves the results in ordinary files, making it easy to analyze the data with conventional tools. tcpflow does have limitations; it does not understand IP fragments or 802.11 headers.

For more details, please see the tcpflow Manual at: http://www.circlemud.org/jelson/software/tcpflow/tcpflow.1.html

A common use of tcpflow is to reveal the contents of HTTP sessions. Using tcpflow you can reconstruct web pages downloaded over HTTP. You can even extract malware delivered as 'drive-by downloads.' (Add graphic)

Lesson 2

Web Traffic Analysis

During an investigation, web-based internet traffic is almost certain to be encountered. The web is used to transfer static and dynamic information, initiate music and video streams, perform file transfers, facilitate online purchases, and act as a conduit through which to attack computers with viruses and malware. The ability to understand web traffic through analysis is a crucial part of any network investigation.

In this lesson, students use Wireshark and NetWitness Investigator tools to analyze HTTP, a commonly used web protocol.

OBJECTIVES *After completing this lesson, students will be able to:*
Identify web clients and web server requests
Classify web traffic by bandwidth used
Perform the extraction of web browser objects/files

HTTP Analysis

HTTP analysis can be approached at different levels. This exercise will start at a high level using NetWitness and progress to the packet level in Wireshark. In this way, the strengths of these two tools can be demonstrated.

Procedure: Opening a Capture File in NetWitness Investigator and Wireshark

Follow these steps to open a capture file simultaneously in NetWitness and Wireshark.

Step	Action
1	Click the NetWitness Investigator icon on the desktop.
2	Click Collection on the menu bar. Then click New Local Collection. Enter a name in the Collection Name field. Leave the path below the Collection Name as Netwitness\Investigations under the user Documents directory. Click OK.
3	The new collection appears in the left column with the name just entered. To the right of the name will be a -. Double click the collection name, and the - should change to Ready.
4	Right click the collection name, and select Import Packets. An Open dialog box will appear. Follow the instructions to locate the desired pcap file to import.During the import process, a progress bar appears in the left column.
5	Double click the Wireshark icon on the desktop.
6	Click Open in the Files section of the opening screen. Select the same file you imported into NetWitness. **Note:** The file may take some time to load.

Procedure: Analyzing a Capture File in NetWitness Investigator and Wireshark

Follow these steps to analyze a capture file in NetWitness Investigator and Wireshark.

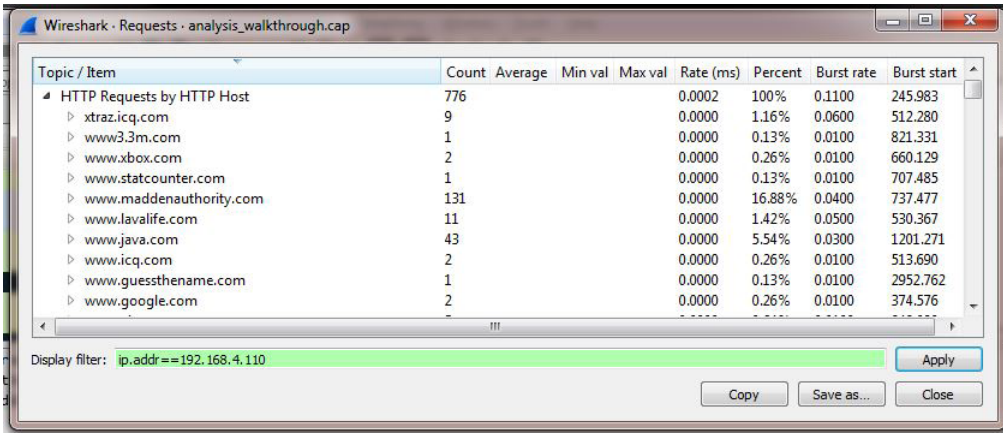
Step	Action
1	In NetWitness, double click the collection name to open a tab to review that collection.
2	Scroll through the upper-level categories to the Hostname Aliases. Each host name is contained in the capture file.
3	Click www.maddenauthority.com. This screen shows that only destination IP address 66.152.98.202 is associated with this site, and that the user feverwilly is referenced in this traffic. The HTTP service type shows 131 sessions in this filter. Click the 131 to the right of HTTP.

Step	Action
4	The sessions are listed in the upper pane. Look at the events to see the filename and directory. Click View under the session screen. Then click txt (View Text) when the session details are displayed. This is POST with the text returned to the server at the bottom, containing the username and password.
5	Click the world icon to View Web in the details menu bar. Click the down arrow in the session pane to advance to the next session. An Authorization Required message is displayed.
6	Click to view the next session. A Web page for Tips, Hints, etc., is displayed. If you display the following sessions, View Web will display the icons and graphics downloaded in each of those sessions.
7	Scroll up to the first session. Click the View Packets icon to the left of "hex". Under packet 1, to the right of the timestamp, is "(63177)".
8	In Wireshark, click Go on the menu bar, and then click Go to Packet. Type 63177 in the Packet Number field, and then click Jump to.
9	Theoretically, this is the first SYN, for the first three-way handshake between these two hosts – 192.168.4.110 and 66.152.98.202. Packets 63181 and 63182 contain the rest of the three-way handshake; 63183 is the POST we looked at in Step 7.
10	Click packet 63183. At the bottom of the packet bytes frame, the username and password of feverwilly can be seen.
11	In the display filter window, type: `Ip.addr==192.168.4.110 and ip.addr==66.152.98.202` Click apply. The display will be reset based on the capture filter. Make sure to scroll to the top of the display. Notice that the first packet number listed is 61758. **Note:** This shows the difference between the two tools. NetWitness has more interpretation in the way it displays information.

HTTP Requests Statistics Using Wireshark

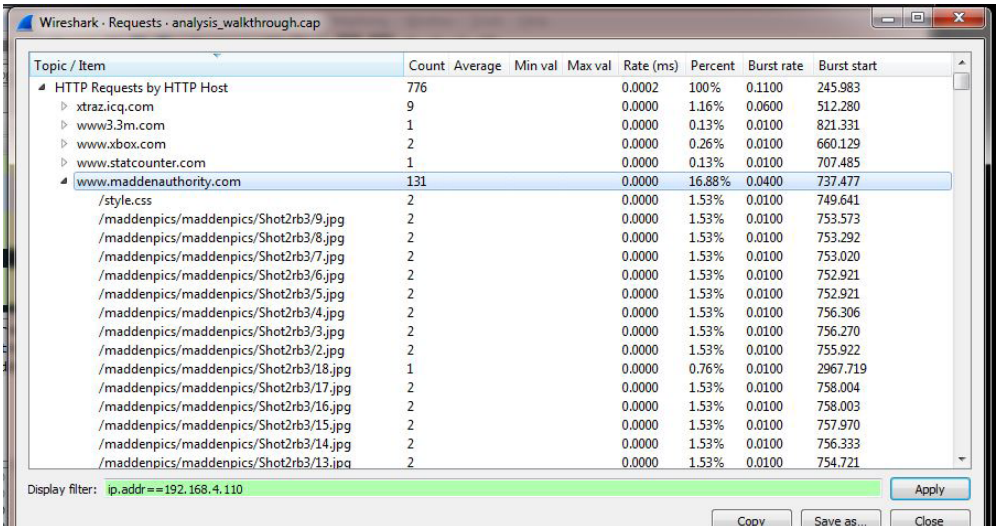
Wireshark can create a custom list of HTTP get requests based on a specified display filter. For example, if the investigator supplies a display filter for a specific IP address, Wireshark shows all get requests for that IP. Creating this statistic requires several steps.

In the image of an HTTP Requests Statistics list, the list includes domain names found in the traffic. The percentage of traffic going to each domain name can also be seen. The example is the result of filtering for all traffic from a single IP address. The picture shows that 131 HTTP requests were made from the filtered IP address to www.maddenauthority.com, the same number seen in NetWitness earlier.



An example of an HTTP Requests Statistics list in Wireshark.

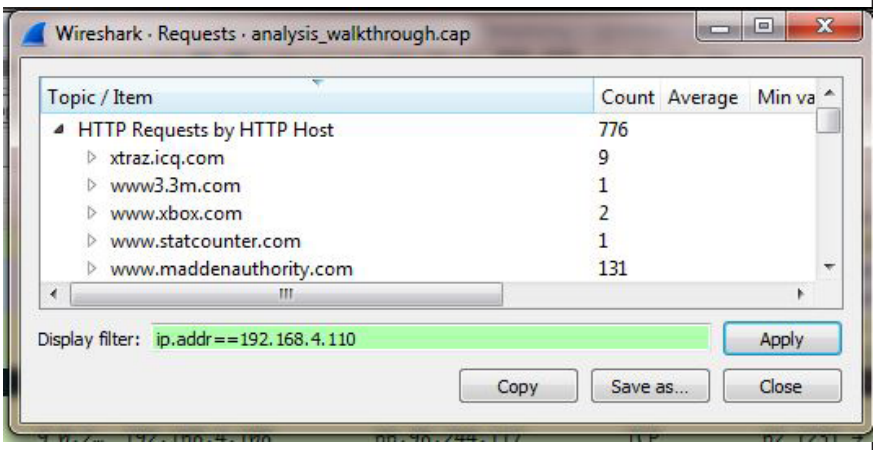
Click the expansion arrow by each domain name to expand a full list of resources accessed from that domain name.



The expansion arrow by www.maddenauthority.com was clicked. Multiple URLs at that domain name were accessed by the computer that was the subject of the filter.

Procedure: Creating a Custom List of HTTP Requests

Follow these steps to create a custom list of HTTP get requests based on a specified display filter.

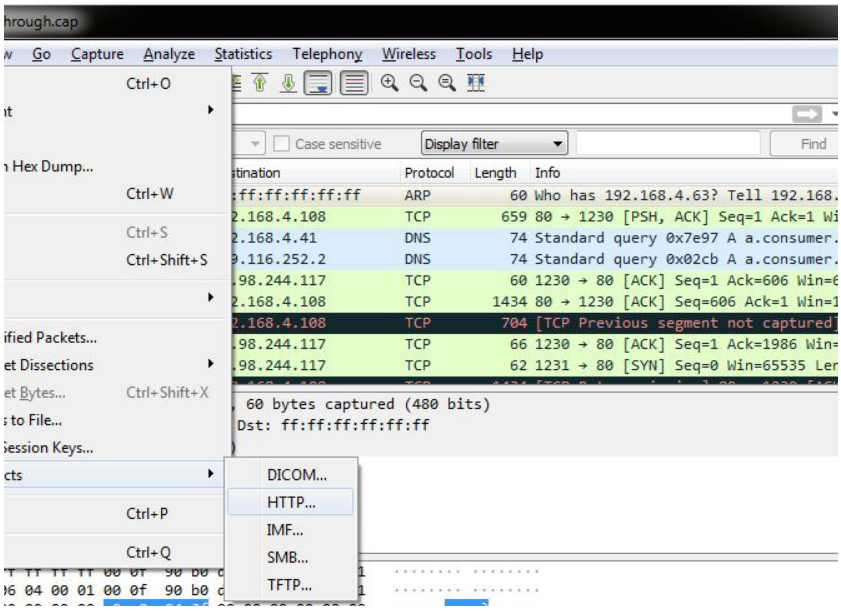
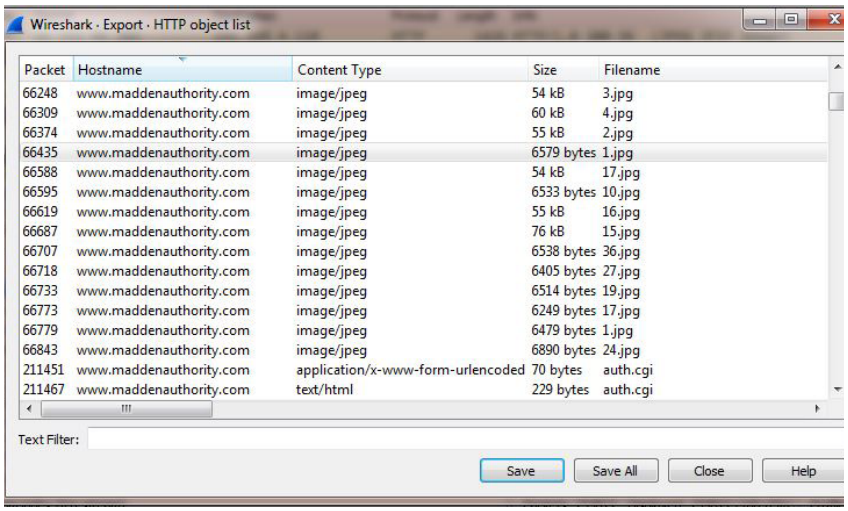
Step	Action
1	Select "HTTP" from the "Statistics" menu in the menu bar.
2	From the menu that displays, select "Requests…" A new window will appear: 
3	Click Apply.

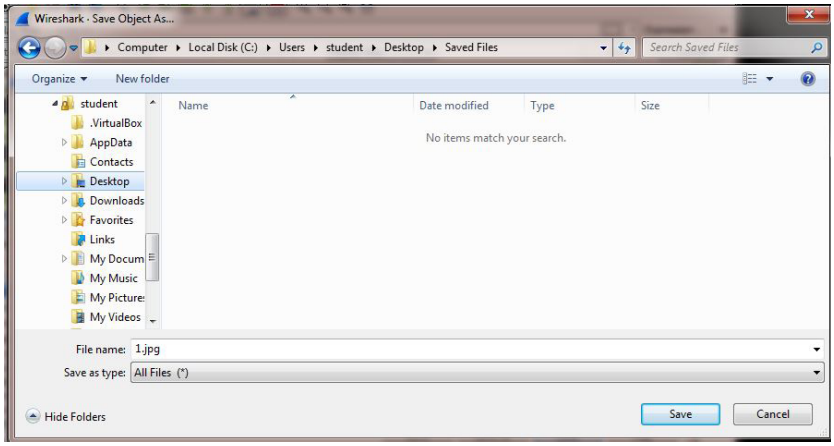
Exporting HTTP Objects Using Wireshark

Wireshark provides a built-in utility that can filter a network capture and identify and list all items transferred during any Web browsing session. Once identified, investigators can export all discovered HTTP objects or a single object with Wireshark. While this utility does not reconstruct a particular Web page, it does allow investigators to determine what type of information was being viewed and identify the IP each object was transferred to or from.

Procedure: Exporting HTTP Objects Using Wireshark

Follow these steps to export an HTTP object with Wireshark.

Step	Action
1	Open the capture file in Wireshark.
2	Click File.
3	Select Export Objects from the drop-down menu.
4	Click HTTP as illustrated in this screen. 
5	You are now presented with the Wireshark HTTP object list.  One object or all objects can be saved. The steps below provide instructions for saving a single object.

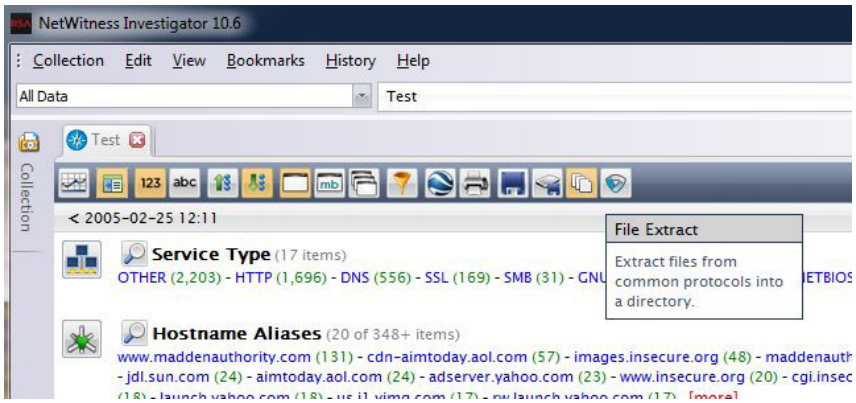
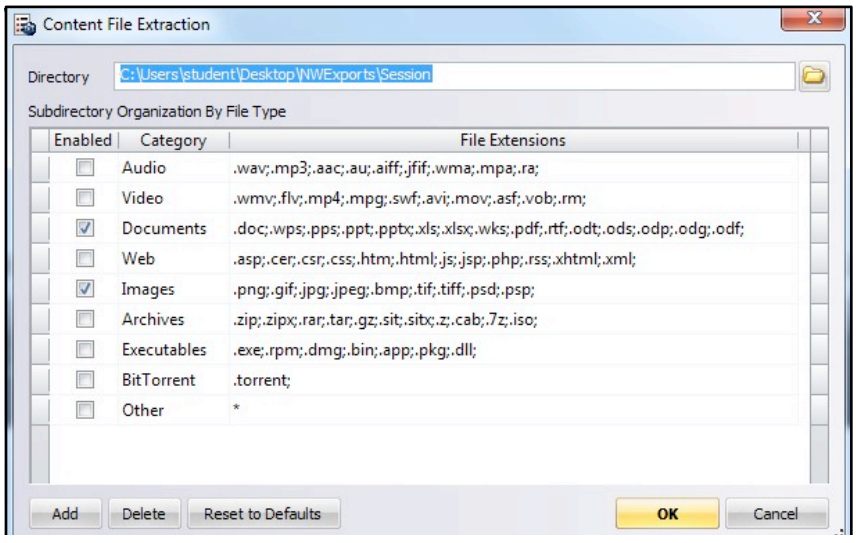
Step	Action
6	Highlight a particular object as shown above and Click Save. The Save Object dialog will be displayed (as shown here). 
7	Use the default folder shown or browse to your preferred folder. Accept the name provided or rename the file and click Save.
8	The saved object can then be viewed by navigating to the location of the saved object and using an application appropriate for the file type to open and view it.

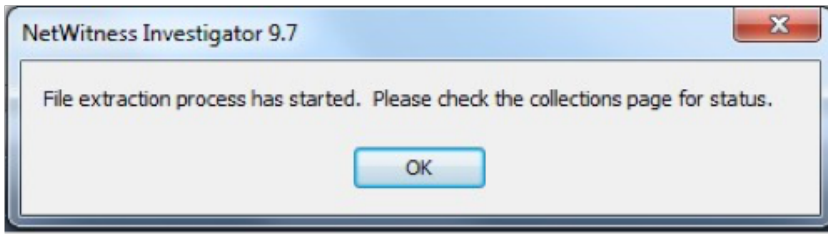
Exporting Files From a NetWitness Investigator Drill View

Like Wireshark, NetWitness Investigator can export files. However, unlike Wireshark, NetWitness Investigator is less restrictive in the files types it will allow the user to export. With NetWitness Investigator, a user may export, in addition to HTTP objects, any common file type, including audio, video, documents and graphic images.

Procedure: Exporting Files From a NetWitness Investigator Drill View

Follow these steps to export a file from a NetWitness Investigator Drill View.

Step	Action
1	While in the top-level view, click the File Extract icon at the top of the pane.  The Content File Extraction dialog will be displayed. Use the default folder shown or browse to your preferred folder. Select the categories of file types to be saved.
2	

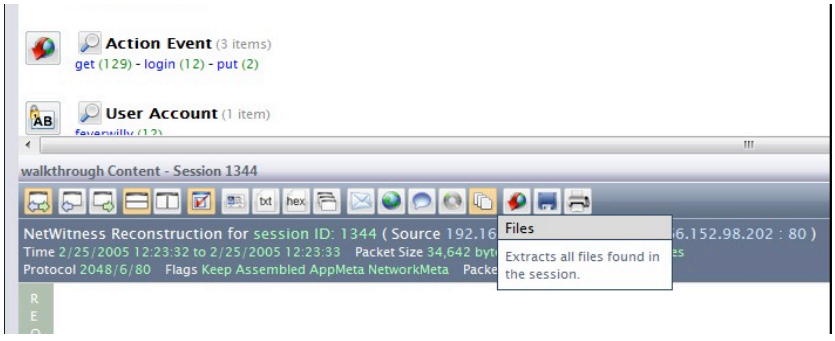
Step	Action
3	Click OK. This dialog box will be displayed: 
4	Click OK.

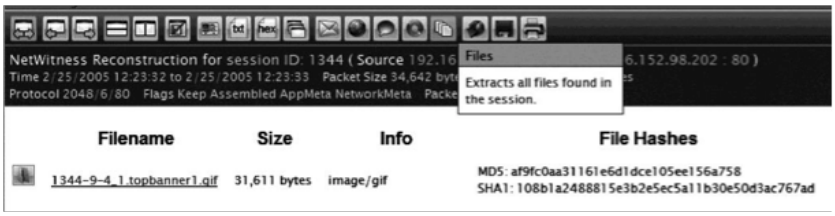
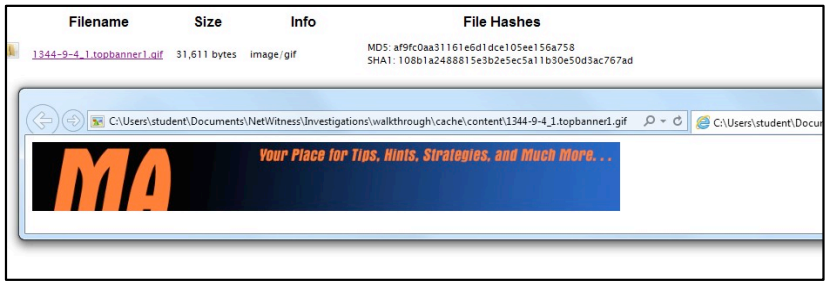
Exporting Files From a NetWitness Investigator Session

The process to export files from a NetWitness Investigator session is similar to the one that allows single Web objects to be exported. The objects can be reviewed before exporting. This allows the export process to be more selective than exporting all objects at one time.

Procedure: Exporting Files From a NetWitness Investigator Session

Follow these steps to export files from a NetWitness Investigator session.

Step	Action
1	While in sessions view, click the Files icon at the top of the session pane. 

Step	Action
2	The list of extracted files from the session will be displayed. 
3	Click one of the listed files. The Open Content dialog will be displayed. Select a method to use to display the content. The default application is usually the best choice. The last option is to save the file in a desired location.  Click OK.
4	The file will be displayed. Here, the gif file is displayed in the IE browser. 

Lesson 3

File Transfer Analysis

Before the development of the web as we know it today, a large part of Internet traffic was the simple transfer of files. The main protocols for file transfers are File Transfer Protocol (FTP) and Server Message Block (SMB). This lesson looks at identifying file transfers and recovering filenames and file content from network capture files.

In this lesson, students use software tools to identify, analyze and recover network transferred files from capture files.

OBJECTIVES *After completing this lesson, students will be able to:*
Locate FTP and SMB file transfers in a capture file
Analyze FTP and SMB file transfers using Wireshark and NetWitness Investigator
Reconstruct file content by exporting appropriate data from capture files

Server Message Block

The Server Message Block (SMB) protocol was developed by Microsoft to transfer files to and from Windows-created shares. The protocol is sometimes called the Common Internet File System (CIFS). The protocol provides for gathering file information, as well as getting and putting files. SMB/CIFS has been ported to other operating systems, most notably Linux.

Both Wireshark and NetWitness Investigator label these file operations as SMB. First, students will analyze SMB file transfers with Wireshark. Then, students will use NetWitness Investigator to do the same analysis.

Detecting SMB File Transfers Using Wireshark

Wireshark can identify the SMB and CIFS file transfer protocols. Packets using this protocol are labeled as SMB in the protocol column. Multiple methods exist of identifying SMB packets in a network capture file:

- Manually scroll through the file looking for occurrences of SMB in the protocol column
- Use the search function (Find Packet under the Edit menu) to execute a display filter search for SMB
- Type SMB directly into the display filter field

Procedure: Detecting SMB File Transfers in a Network Capture File

Follow these steps to determine the presence of SMB packets in the capture file.

Step	Action
1	Double click the Wireshark icon on the desktop.
2	From the opening screen, click the files section, and then click open. Select a file as directed by the instructor. The file may take some time to load.
3	Click Edit on the menu bar. Then click Find Packet. For this search, leave it as Display filter. Type smb in the filter field. Click Find.
4	A packet meeting the smb search criteria will be highlighted. Typing Ctrl-N will search for the next occurrence and highlight that packet.
5	Typing smb in the Filter field creates a display filter for SMB protocol packets.
6	Any packets found indicate SMB activity. Read or write commands indicate attempts to transfer files.

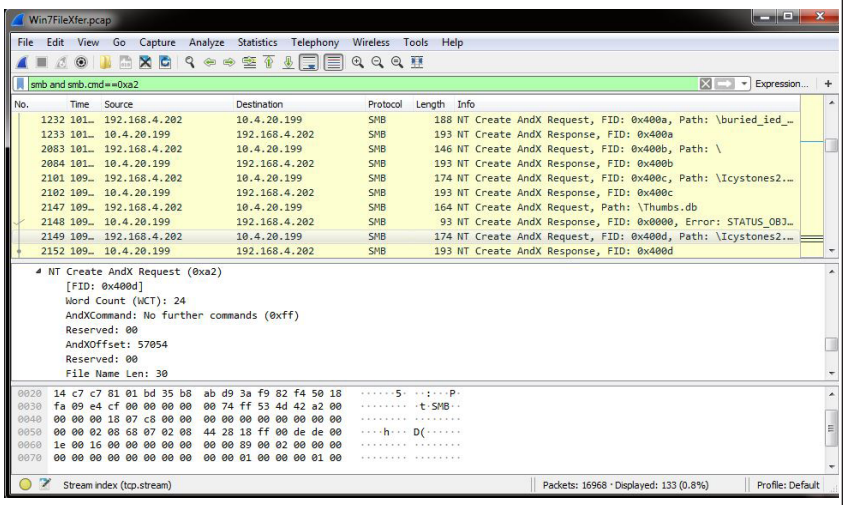
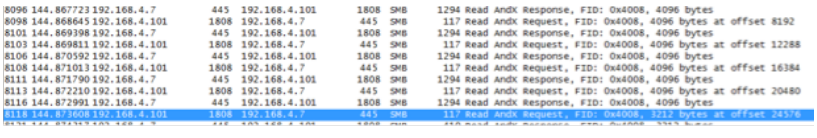
Investigating SMB Activity Using Wireshark

If SMB activity exists, gathering details might be important. Some details are the source and destination computers (IP addresses), names and sizes of files involved, timestamp of the files and direction of the file transfer.

The SMB protocol associates a filename with a file ID (FID), a four-digit hexadecimal number. After discovering the FID through file association, activity can be traced through the FID.

Procedure: Investigating SMB Activity Details Using Wireshark

Follow these steps to investigate SMB activity details using Wireshark.

Step	Action
1	Use the file already open in Wireshark. Create a display filter for SMB, if desired.
2	Look for packets that associate a FID with a filename. A helpful display filter is: smb and smb.cmd == 0xa2
3	Match that FID with a Create Andx Response. In the middle Wireshark pane, expand SMB, then expand SMB Header, and then expand NT Create Andx Response. If the file exists, the time and date stamps, size and filename will be shown. 
4	Use this display filter to create a display showing only SMB reads (0x2e) and writes (0x2f): `smb.cmd == 0x2e or smb.cmd == 0x2f` The middle pane contains the filename under the SMB Header information.
5	A complete in-order file transfer looks like this: 
6	The above methodology can be used to identify all attempted file transfers and indicate the likely successful transfers.

Reconstructing File Content of SMB File Transfers Using Wireshark

The content of the transferred files should be in the network capture file. Reconstructing these files may be a necessary part of an investigation. Filenames can be changed, and verifying the content of the files transferred might be important. Wireshark provides a menu-driven function to identify file transfers and extract the files from the network capture.

The SMB Object List shows all the file transfers found in the network capture file. For each transfer, this information is listed: packet number, host name, content type and assumed percentage complete, exported file size, and exported filename. All items in the list can be exported at once using the Save All button, or one item can be highlighted and exported using the Save As button. The export location can be chosen in the Save Objects In dialog.

Procedure: Exporting SMB Objects Using Wireshark

Follow these steps to reconstruct and export files by name.

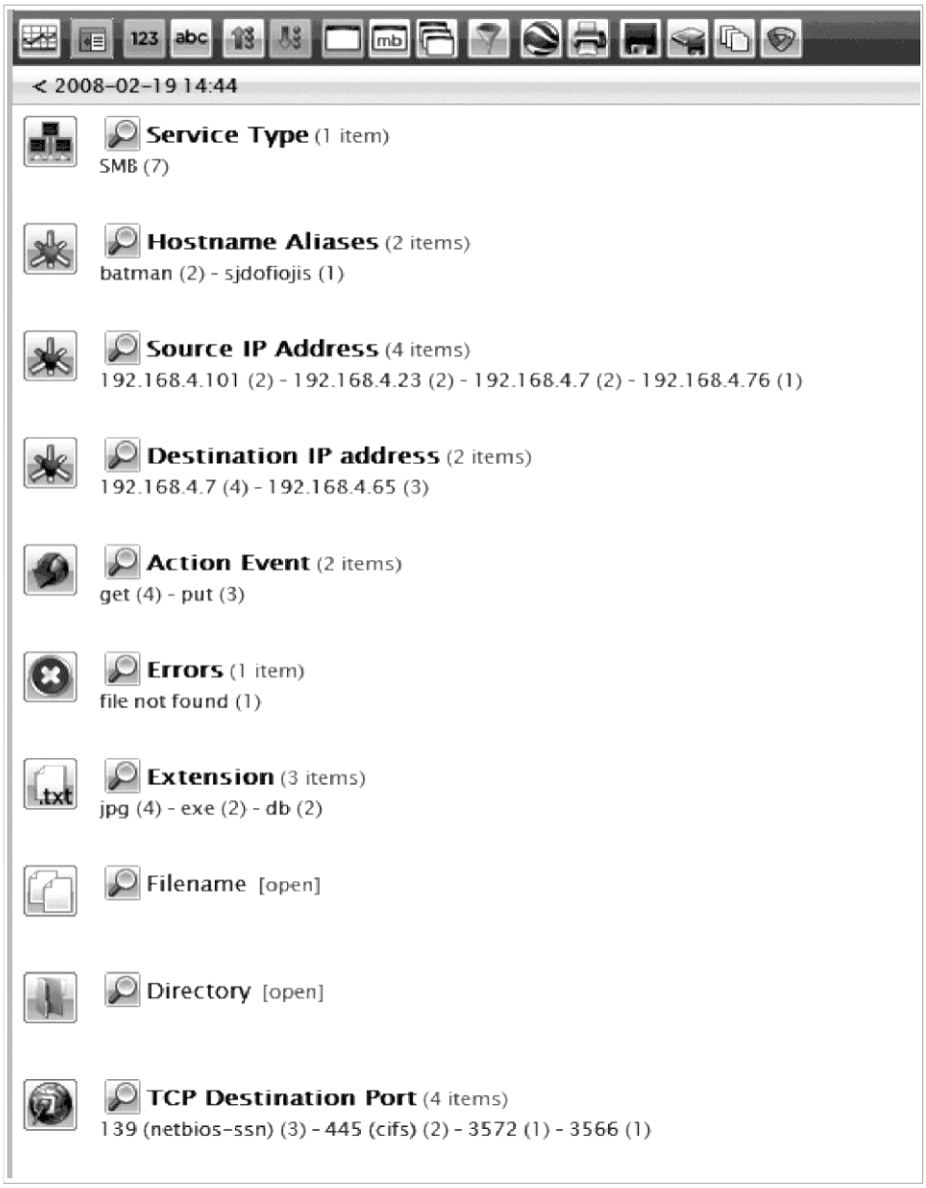
Step	Action
1	Use the file already open in Wireshark.
2	Click File in the menu.
3	Click or hover over Export Objects. Then click SMB. The SMB Object List will be displayed in a new window.
4	Highlight a single object in the SMB Object List if desired. The percent complete value may be useful in making this decision. Click the Save button for one object, or the Save All button for the entire list.
5	Navigate to the desired directory in the Save Objects dialog. Enter a filename or accept the default name. Then click Save. If you clicked Save All, enter the name of a directory to be created to store all exported objects. Then click OK.
6	The export will take place. The exported files can be reviewed using Windows Explorer.

Detecting SMB File Transfers Using NetWitness Investigator

NetWitness Investigator can identify the SMB and CIFS file transfer protocols. Sessions using this protocol are automatically added to the category name SMB.

When looking at the top level of a collection, click the word SMB to filter the top-level display so that only SMB sessions remain on the top-level display. All appropriate categories still contain data about the SMB sessions, as shown here.

NetWitness Investigator displays SMB session data.



Procedure: Detecting SMB File Transfers in a Network Capture File

Follow these steps to determine the presence of SMB packets in the capture file.

Step	Action
1	Click the NetWitness Investigator icon on the desktop.
2	Click Collection on the menu bar. Then click New Local Collection. Enter a name in the Collection Name field. Leave the path below the Collection Name as Netwitness\Investigations under the user Documents directory. Click OK.
3	Notice that the new collection appears in the left hand column with the name just entered. To the right of the name there will be a -. Double click the collection name and the - should change to Ready.
4	Right click the collection name and select Import Packets. An Open dialog box will appear. Follow the instructions given to locate the desired pcap file to import. During the import process, there will be a progress bar shown in the left column.
5	Double click the collection name to open up a tab to review that collection.
6	Note the upper level categories. If SMB is listed under Service Types with a number in green next to it, there are SMB sessions in the collection.

Investigating SMB Activity Using NetWitness Investigator

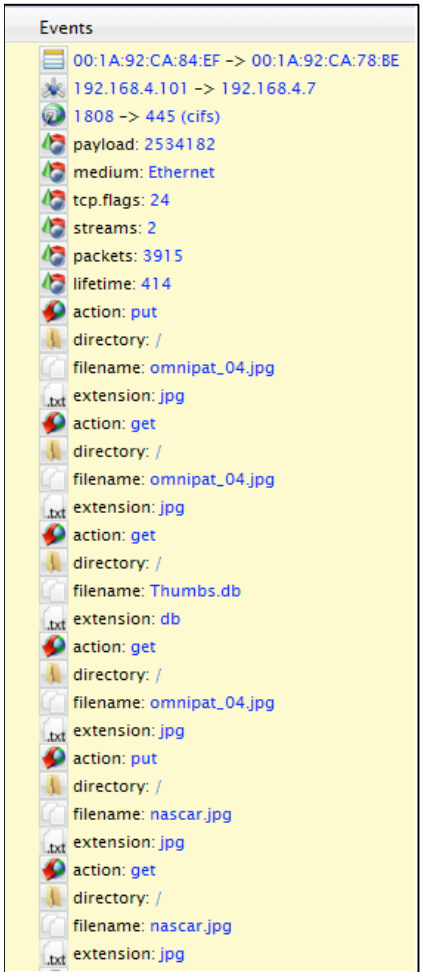
When looking at the top level of a collection, click the number in parentheses to the right of the word SMB to enter the sessions view directly. Only the sessions pertaining to SMB activity will be listed. On the right side, the activity will be shown. The filenames and direction of transfer (get or put) are of interest.

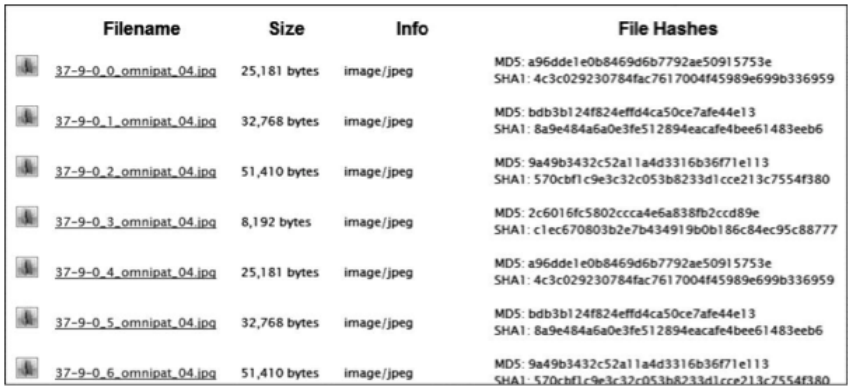
NetWitness Investigator does not provide the ability to easily associate filenames with FIDs, or to find the MAC times of the transferred files. The Session Files view lists the extracted files and their sizes.

One benefit of NetWitness Investigator is that the content of each file can be displayed from within the Session view.

Procedure: Investigating SMB Activity Using NetWitness Investigator

Follow these steps to investigate SMB activity using NetWitness Investigator.

Step	Action
1	Start at the top level of the collection used in the last procedure. The top-level display should fill the screen.
2	Click SMB under Service Type to drill down into SMB, but remain at the top-level display. This is filtering the top-level display for SMB sessions.
3	This display shows the source and destination IP addresses, and the types of files are listed under Extensions.
4	Click the green number to the right of SMB to open the sessions view. Filenames should be listed under the events column. 

Step	Action
5	Click View under one of the sessions. Click Files at the top of the session pane. A list of files reconstructed from the session will be listed. This list may be more extensive than the list in the events column in the step above. The file size is listed, as well as the md5 and SHA1 hash values for the reconstructed file, but no other metadata is listed. 
6	Click View under each of the other sessions. The file list will be displayed for each.
7	The files can be previewed without exporting. Click a filename in the Session File view. An Open Content dialog will be displayed. Under How would you like to view this content?, you can choose Open In Navigator, or Open using the default application for file type. The content will be displayed.

Reconstructing File Content of SMB File Transfers With NetWitness Investigator

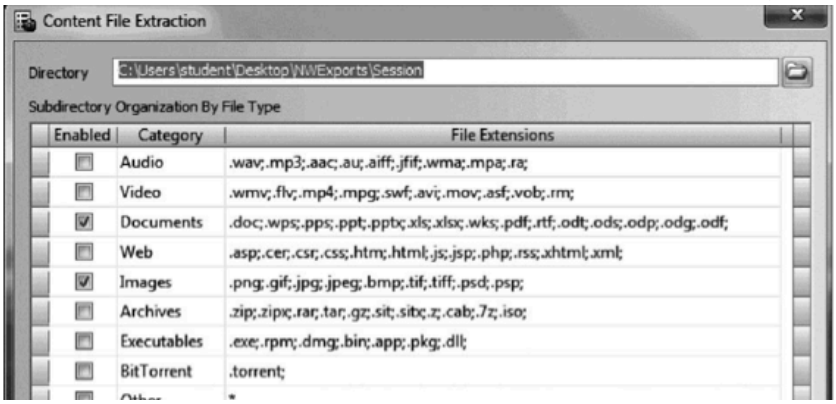
Reconstructing SMB files is easy in NetWitness. The process is similar to the exporting procedures introduced during the Web analysis lesson. Filtering the top-level display for SMB sessions limits the File Extract process to the files in the SMB sessions.

The Files View in the Sessions pane provides the ability to preview and export one file at a time.

Procedure: Exporting All SMB Objects Within a Capture File

Follow these steps to export all the SMB file objects contained in the capture file.

Note: This is accomplished by filtering the top-level view using the SMB filter.

Step	Action
1	Start at the top level of the collection used in the last procedure. The top-level display should fill the screen.
2	Click SMB under Service Type to drill down into SMB, but remain at the top-level display. This is filtering the top-level display for SMB sessions.
3	Click the File Extract icon.  The Content File Extraction dialog will be displayed.
4	Select the desired directory to export the files. Clicking the folder icon to the right of the field opens a file browser. Click OK in the File Browser to populate the directory field.
5	Put check marks in all the enable boxes for the desired categories.  Check all the options to ensure no files get missed.

Procedure: Exporting SMB File Objects From Within Session

Follow these steps to export a file at a time from within the session view.

Step	Action
1	Start at the top level of the collection used in the last procedure. The top-level display should fill the screen.
2	Click the green number to the right of SMB. The Session view will be displayed.
3	Click the view button under one of the session icons. This will open the session pane in the bottom portion of the screen.
4	Click Files at the top of the session pane.
5	A list of files reconstructed from the session will be listed. This list may be more extensive than the list in the events column in the step above. The file size is listed, as well as the md5 and SHA1 hash values for the reconstructed file, but no other metadata is listed. 
6	The files can be exported one at a time. Click a filename in the Session File view. An Open Content dialog will be displayed. Under How would you like to view this content?, choose Save the content to a specific location. Click OK. 
7	The Save As dialog opens. Choose the destination location for the file export. Change the name of the file as desired. Click Save. **Note:** The default filename includes the session information, which may be helpful in later analysis.

File Transfer Protocol

File Transfer Protocol (FTP) is a standard network protocol used to transfer files between computers. FTP uses a client-server architecture, with users running FTP client applications connecting to computers running FTP server applications. The FTP protocol requires a login, but if anonymous login is enabled, no validation of the login credentials exists.

FTP typically uses two sessions during operation. The first session is the command session. The second session is used to execute the actual data transfers. The command session typically uses server port 21, and the data session typically uses server port 20, but this is easily changed.

In FTP, the command and data transfers are in cleartext. The SSH File Transfer Protocol (SFTP) encrypts both the command and data sessions. Another file transfer protocol, Trivial File Transfer Protocol (TFTP), uses UDP port 69 to initiate file transfers without any authentication or encryption. This discussion is limited to FTP, but some of the concepts apply to other file transfer protocols.

Detecting FTP File Transfers Using Wireshark

As with SMB, Wireshark can identify FTP protocols. Packets using this protocol are labeled FTP or FTP-DATA in the protocol column. These labels can be identified using the same techniques described under SMB: manual inspection, searching, and using display filters.

Procedure: Detecting FTP File Transfers in a Network Capture File

Follow these steps to determine the presence of FTP packets in the capture file.

Step	Action
1	Open the Wireshark application.
2	The main window will have files listed under the Open section. Select a file as directed by the instructor.
3	Click Edit on the menu bar. Then click Find Packet. For this search, leave it as Display filter. Type ftp or ftp-data in the filter field to search for command and data packets. Click Find. **Note:** The search term can be either ftp or ftp-data if the user wants to search for only one of these packet types.
4	A packet meeting the search criteria will be highlighted. Typing Ctrl-N will search for the next occurrence and highlight that packet.
5	Typing ftp or ftp-data in the Filter field creates a display filter for FTP command and data transfer packets. **Note:** As above, the search term can be either ftp or ftp-data as desired.
6	Any packets found indicate FTP activity.

Investigating FTP Activity Using Wireshark

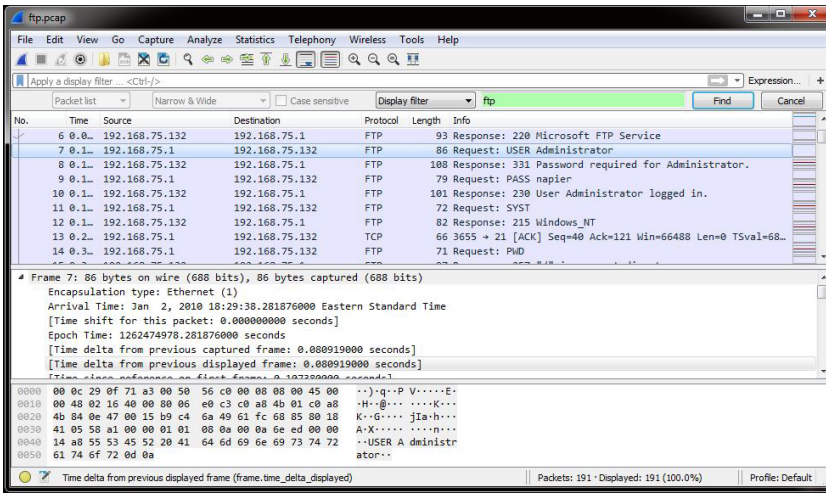
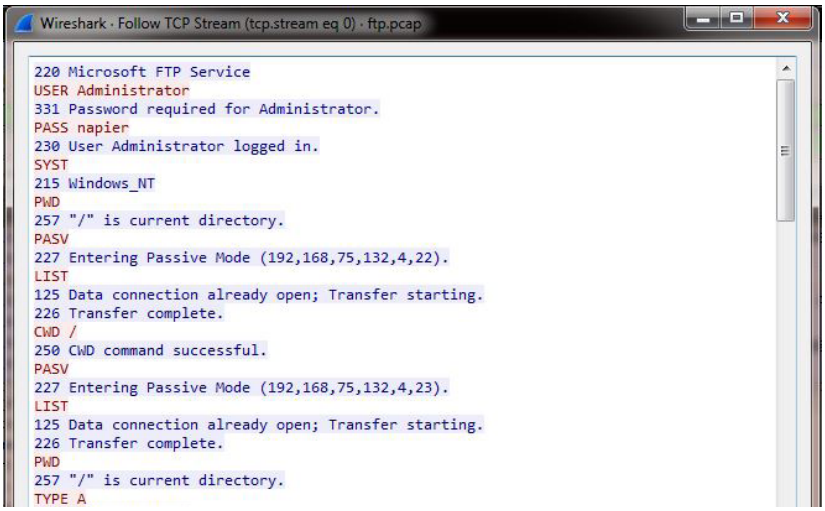
If FTP file transfers are taking place, gathering details might be important. Some details are the source and destination computers (IP addresses), names of files involved, the sizes of these files, and the direction of the transfer.

FTP can be used to gather intelligence about files on an FTP server, so executing a directory listing against server directories could be important information.

When using a text-based ftp client, the get and put commands are used to transfer files from and to the server, respectively. The actual commands transmitted to the server are different. The RETR command requests a file be returned to the client, and the STOR command transmits a file to the server. These are the commands to look for in the network capture.

Procedure: Investigating FTP Activity Details Using Wireshark

Follow these steps to determine FTP activity details using Wireshark.

Step	Action
1	Use the file already open in Wireshark. Create a display filter for ftp. Include ftp-data in the display filter if desired.
2	Identify the IP addresses of the client and server.
3	Identify the login credential s used to access the server. 
4	To identify the commands during the session, highlight an FTP protocol line in the top window. Right click that line, and then select Follow TCP Stream. 
5	The ftp display filter facilitates the discovery of all ftp clients and servers in this capture. The Follow TCP Stream output can be used to identify credentials, attempted file transfers and directory enumeration.

Reconstructing File Content of FTP Transfers Using Wireshark

The content of the transferred files should be in the network capture file. Reconstructing these files may be a necessary part of an investigation. Filenames can be changed, and verifying the content of the files transferred might be important. Since FTP data transfers are done in separate sessions, exporting the content of a session during which a file was transferred provides the file content.

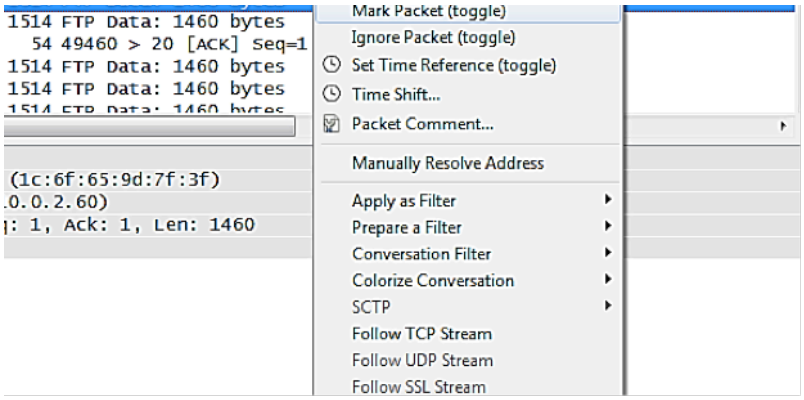
The data transfer session follows the command for the data. After the packet from the client to the command port containing the request, a three-way handshake will establish the data transfer session. The command packets can be found using the search function by creating a search string containing the ftp.request.command keyword. Look for values such as LIST (for a directory listing), RETR (for requesting a file from the server) and STOR (for sending a file to the server). The complete search string might look like this:

```
ftp.request.command == “LIST” || ftp.request.command ==
“RETR” || ftp.request.command == “STOR”
```

By “Following the TCP Stream” that is initiated for the data transfer, the file content, or directory listing, is extracted. Once extracted from the capture file, the file can be exported.

Procedure: Exporting FTP Files Using Wireshark

Follow these steps to extract and export files by name.

Step	Action
1	Use the file already open in Wireshark.
2	Clear any active display filter. Click Edit, and then click Find Packet.
3	The Find Packet dialog opens. Select the Display Filter radio button. In the Filter field, type: `ftp.request.command == "RETR" || ftp.request.command == "STOR"` Note: This is one line of typing. Click Find.
4	The dialog disappears, and the first instance of a file transfer command is highlighted. Note the filename in the STOR or RETR command.
5	Highlight a packet in the three-way handshake, or an ftp-data packet, between the client and the server. Right click, and click Follow TCP Stream. 
6	The Follow TCP Stream window that pops up contains the content of the file. Select the Raw in the drop down beside "Show and save data as" under the data window.
7	The Save Follow Stream Dialog window appears. Select a folder to store the file in and name the file as desired. Click Save. The file will be saved and the window closes.
8	Click Close in the Follow TCP Stream window.
9	Search for the next file transfer instance by pressing Ctrl-N. Repeat steps 4 through 8 to save each file as desired.

Detecting FTP File Transfers Using NetWitness Investigator

NetWitness Investigator can identify the FTP protocol. Sessions using this protocol are automatically added to the category name FTP under Service Types.

When looking at the top level of a collection, click the word FTP to filter the top-level display such that only FTP sessions remain on the top-level display. All appropriate categories still contain data about the FTP sessions. This is demonstrated in the next procedure.

Procedure: Detecting FTP File Transfers in a Network Capture File

Follow these steps to determine the presence of FTP packets in the capture file using NetWitness Investigator.

Step	Action
1	Click the NetWitness Investigator icon on the desktop.
2	Click Collection on the menu bar. Then click New Local Collection. Enter a name in the Collection Name field. Leave the path below the Collection Name as Netwitness\Investigations under the user Documents directory. Click OK. Notice that the new collection appears in the left column with the name just entered. To the right of the name will be a -. Double click the collection name, and the - should change to Ready.
3	Right click the collection name, and then select Import Packets. An Open dialog box will appear. Follow the instructions to locate the desired pcap file to import. During the import process, a progress bar appears in the left column.
4	Double click the collection name to open a tab to review that collection.
5	Note the upper-level categories. If FTP is listed under Service Types with a number in green next to it, FTP sessions are in the collection.

Investigating FTP Transfers Using NetWitness Investigator

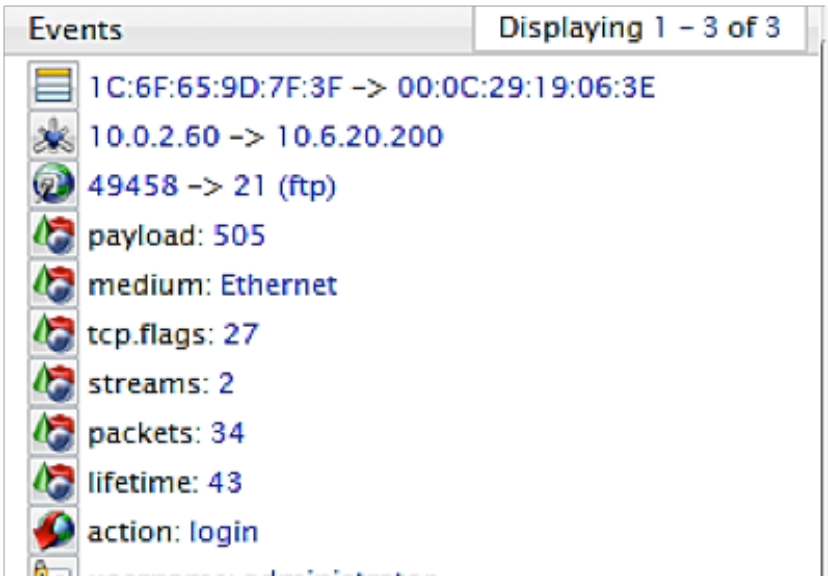
When looking at the top level of a collection, clicking on FTP will filter the top-level view. In this filtered view, FTP client and server IP addresses are displayed, along with user accounts and other category information. Click the number in parentheses to the right of the word FTP to enter the sessions view directly. Only the sessions pertaining to FTP activity will be listed. On the right side, the activity will be shown. The filenames and direction of transfer (get or put) are of interest.

NetWitness Investigator does not provide the ability to identify the MAC times of the transferred files. The Session Files view lists the extracted files and their sizes.

One benefit of NetWitness Investigator is that the content of each file can be displayed from within the Session view.

Procedure: Investigating FTP Activity Using NetWitness Investigator

Follow these steps to Investigate FTP activity using NetWitness Investigator.

Step	Action
1	Start at the top level of the collection used in the last procedure. The top-level display should fill the screen.
2	Click FTP under Service Type to drill down into FTP, but remain at the top-level display. This is filtering the top-level display for FTP sessions.
3	This display shows the source and destination IP addresses, User Accounts and action events (get/put).
4	Click the green number to the right of FTP to open the sessions view. Filenames should be listed under the events column next to each session. 
5	Scrolling through the sessions listed, notice the FTP command sessions and FTP data sessions. These may be identified by the port number and/or ftp vice ftp-data.

Step	Action
6	Click View under the FTP command session; the destination port should be 21 (ftp). Click txt at the top of the session pane. The bottom window displays two columns – request on the left and response on the right. This display shows all the commands and responses that make up this FTP session. Evaluate this information to determine how many data sessions are present and which one represents file transfers.
7	Click Files at the top of the session pane. A file with a .raw extension will be listed and can be opened for viewing. This file contains both sides of the dialog in the command session between the server and client. This file can be exported as desired. **Note:** This step may open a second tab in the upper window. Close this tab before proceeding to the next step.
8	In the upper window, look at the data sessions. One of the entries in the Events column is payload (number of bytes of data transferred). Large numbers indicate a file transfer. Smaller numbers may be small files or directory listings.
9	In the upper window, click View under a data session. Click txt at the top of the session pane. The bottom window contains data in the column at right. This is the text representation of the data transferred. For a LIST command, it will be a directory listing. For a RETR or STOR command, it will be the file content, which may be unintelligible.
10	Review all command sessions. This is where the account and files transferred information can be found.

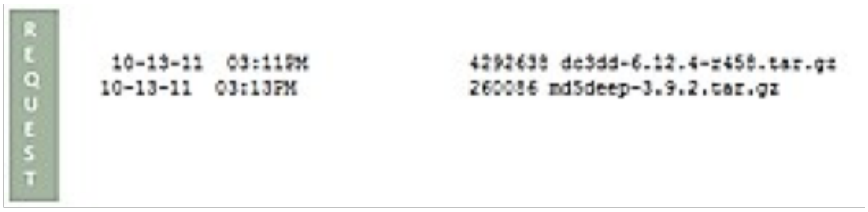
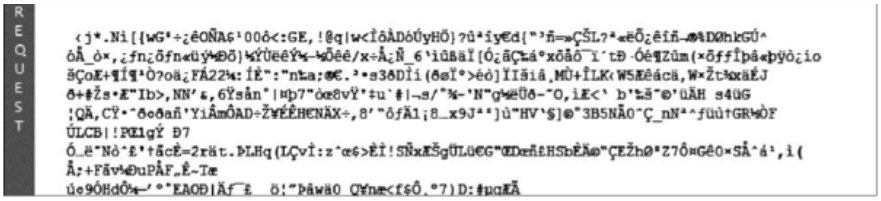
Reconstructing File Content of FTP Transfers Using NetWitness Investigator

Reconstructing FTP files is similar to the exporting procedures for individual files introduced during the Web analysis lesson earlier.

NetWitness cannot associate ftp-data sessions with the original filenames; therefore, files cannot be exported using the File Extract function. FTP opens a separate session for each data transfer. The Files View in the Sessions pane provides the ability to export this file.

Procedure: Exporting an FTP File From Within a Session

Follow these steps to export a file at a time from within the session view.

Step	Action
1	Start at the top level of the collection used in the last procedure. The top-level display should fill the screen.
2	Click the green number to the right of FTP. The Session view will be displayed.
3	In the upper window, click View under a data session for a file transfer. Click the txt icon at the top of the sessions view. By examining the left column, you should determine if this session is a file transfer or a directory listing. 
4	View a file transfer data session. 
5	With Files selected at the top of the session pane, a file will be listed with a .raw extension. The md5 and SHA1 hash values for the file are also listed. This .raw file is the content of the transferred file. The hash values should match those of the original file. Note: Recalling how FTP works, a new session is opened for each data transfer. Therefore, only one file will be listed here.

Step	Action
6	To export the file, click the filename in the Session File view. The Open Content dialog will be displayed. Under "How would you like to view this content?", choose "Save to a specific location". Click OK. 
7	The Save As dialog opens. Choose the destination location for the file export. Change the name of the file as desired. Click Save. Note: The default filename includes the session information, which may be helpful in later analysis.

Lesson 4

Introduction to Intrusion Analysis

When analyzing the capture of a network monitoring session, reviewing the data capture for indicators of a network intrusion is prudent. The Snort intrusion detection software is one method to analyze a capture and identify indicators of a network intrusion.

In this lesson, students learn to configure and use a software-based intrusion detection system to monitor and analyze a network for indicators of a network intrusion.

OBJECTIVES *After completing this lesson, students will be able to:*
Define the concepts of rule-based network traffic inspection
Configure network detection rules
Interpret network event alerts
Obtain packet traffic based on network events
Identify intrusion artifacts of some common attacks

Intrusion Detection

Intrusions can originate from inside or outside a network or network segment and represent anomalous activity that may go against that network's policies or represent illegal activities. Evidence of an intrusion may directly affect a case, bringing into question the integrity of the network and the investigation. So investigators must be familiar with and be able to quickly identify activity of this type to help ensure the integrity of their network and their investigation.

Methods of identifying intrusion events vary depending on how the intrusion was performed and how the existing network devices are managed and monitored. As an intrusion traverses witness devices and targeted systems, it can create indicators that point to its existence. An investigator's ability to spot these indicators is vital to identifying an intrusion. An intrusion can remain undetected indefinitely if no one is looking for it, or if indications of its existence are not recorded or do not activate an alert.

Typically, intrusions are first identified by a security device on the network configured to alert when a certain condition is witnessed. Intrusion detection or prevention systems (IDS/IPS), antivirus software (A/V), network proxies and log management devices are designed to monitor for specific signatures, headers or traffic behavior and alert when they present. These alerts are indicators that suggest aberrant network activity; they do not necessarily mean an intrusion event has occurred. Further investigation and analysis are usually necessary to uncover more indicators and perhaps pinpoint and define an intrusion event.

IDSs and IPSs, like A/V software, are only as good as the signatures and rules they use to detect aberrant network indicators. If no rule or signature exists for a particular type of intrusion, no alert will be issued, and consequently, the event is unlikely to be noticed. An analyst must proactively look for intrusion indicators in the right places to find them. One of the most effective ways to find indicators of network intrusion events is to know and understand what normal network traffic looks like for the particular network of interest.

When investigators have a comprehensive understanding of "normal" traffic, abnormal traffic tends to stand out.

Indicators of compromise are clues that an intrusion may have occurred. An indicator can take on many forms. It can be as obvious as an alert from a network security device or as ambiguous as a statistically anomalous process within the multitude of legitimate ones running on a network. Understanding the infrastructure, security devices and normal behavior of a network and the systems on it are key to identifying indicators of compromise. A single indicator by itself may mean nothing to a casual observer. Only when it is put into context does it suggest an intrusion. Some common indicators of nefarious network activity can include:

- Antivirus alerts and quarantines
- Logged system and/or application error events
- Unauthorized software on hosts
- Alerts from network management devices
- Unusual, unexpected and/or unauthorized connections
- Uncharacteristic traffic into or out of the network segment

Snort

Snort is a free multi-platform, network packet capture and analysis tool. It can be downloaded from http://www.snort.org/. Snort inspects network traffic for specific signatures, protocols and/or packet content based on preconfigured rule definitions. Snort can monitor and capture live traffic or analyze and inspect previously captured network traffic.

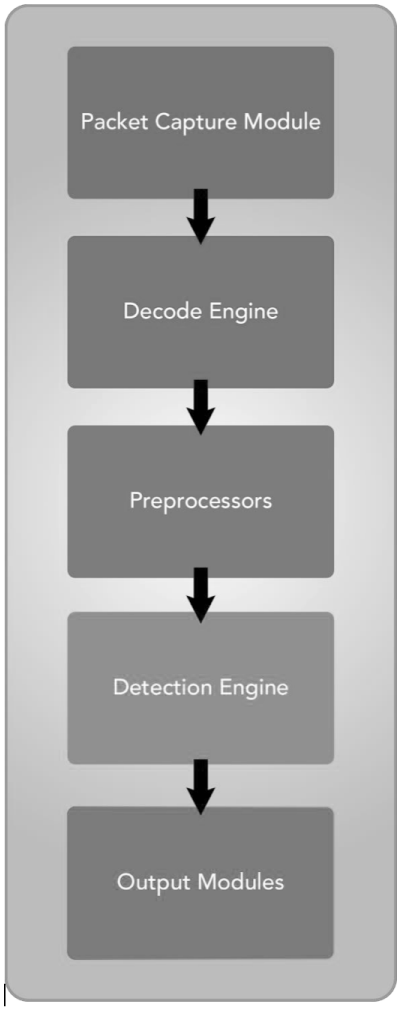
Snort runs in three modes:

- Sniffer mode: Captures all packets across an interface and displays them to the console
- Packet Logger mode: Captures all packets (filtered or not) across an interface and saves them to a storage device
- Network Intrusion Detection (NIDS) mode: Captures only packets defined by a rule set and saves them to a storage device

Application Components

Snort includes the following major application components. Typically, these are not configured directly, but understanding what these modules do and how data flows through Snort is important for configuration and troubleshooting purposes.

- **Packet Capture Module:** Performs a network packet capture. Uses the libpcap (or winpcap on Windows devices) packet capture library to collect traffic.
- **Decode Engine:** Module responsible for deconstructing protocol headers from TCP/IP layers 1-3. Data from these headers will be recorded and tested for abnormalities.
- **Preprocessors:** Modules that perform a variety of functions, such as reassembling TCP streams and checking to see if application protocols are behaving normally.
- **Detection Engine:** Module that compares captured traffic to the IDS rules to determine if there is a match.
- **Output Plugins:** Modules used to feed Snort output to a variety of sources in a variety of formats, from text files to database entries.



Snort's major application components.

Files and Directories

Several directories are created when Snort is installed. Among others, these are created within the Snort install directory:

- **bin:** Location for executables/libraries
- **rules:** Location of files that contain the IDS rules; any file with a ".rules" extension
- **etc:** Location of configuration files, including snort.conf, the primary configuration file
- **doc:** Location of "readme" files and the Snort manual

Snort.conf

The file snort.conf is the primary location where configuration changes are made to the Snort application. The snort.conf file is in the "etc" directory wherever Snort is installed. Snort.conf is a text file, so changes can be made with a text editor.

In this file, lines that are comments begin with a # symbol. All other lines will be read and used when Snort is run. To configure this file, users must edit a non-comment line, add a new line, or uncomment (remove the #) from a comment line.

Editing this file is used for these tasks, among others:

- Ruleset customization
- Preprocessor activation/deactivation and tuning
- Changing output type
- Changing variables

Configuring Snort

Configuring and tuning an IDS is a complex and ongoing task, and it will not be addressed fully in this text. However, explanations of several configuration changes follow.

Snort.conf: Activating and Deactivating Rule Files

Many rule files are typically active in a running Snort process. Individual rule files can be turned on or off in the snort.conf file. This is useful if a new rule file is created and needs to be added, or if a rule file needs to be turned off to narrow searches. A rule file configuration line in snort.conf looks like this:

```
include $RULE_PATH/file.rules
```

To remove a rule file, add a "#" symbol in front to comment it out. For example, the file named above could be temporarily removed from use by changing the line to:

```
#include $RULE_PATH/file.rules
```

A new file can also be added by adding a new line, with the proper filename at the end. For example, to add the rule file named "case001.rules," add this line to snort.conf:

```
include $RULE_PATH/case001.rules
```

Ensure that the rule file exists and is in the path specified in the rule location variable. By default, $RULE_PATH points to c:\Snort\rules

Defining a Filter for Snort

Snort contains a structured system for creating alerts and for flagging traffic on a network using its rules system. However, Snort also uses the standard tcpdump filters to sort through network traffic. To create a capture filter that will collect all required information without stepping over any legal boundaries, the usage and syntax of these filters must be understood.

To capture all traffic for a single IP address, use:

```
host <ip address>
```

To capture all tcp traffic over a single port, use:

```
tcp port <port number>
```

These values can also be combined for complex filters. For example, to search for all tcp traffic involving 192.168.4.10 going over ports 25 and 110, use this filter:

```
host 192.168.4.10 and (tcp port 25 or tcp port 110)
```

Snort Syntax and Options

```
snort -C -i <network interface> -l <log directory> -L
<log file> -v <filter>
```

Option	Description
-C	Print payload with characters only – not hexadecimal
-i	Specify network interface; based on numbered entries found with snort -W
-l	Directory where log/alert files will be stored
-L	Log/alert filename
-v	Verbose output to screen
-P	Set snap length of packet to capture (default 1514 bytes)
-c	Use specific rules/configuration file

Procedure: Activating Snort

Follow these steps to start Snort and capture traffic related to a specific host.

Step	Action
1	Open a command terminal.

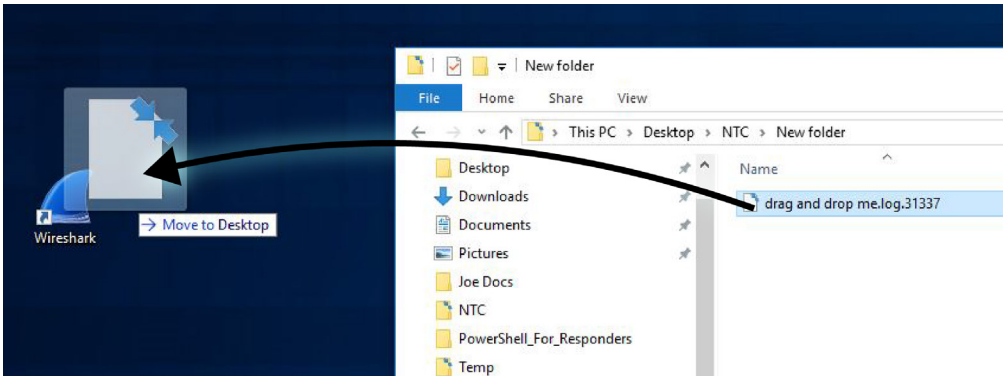
2	At the Windows command line, determine which NIC is your out-of-band monitor. Type:
	
	`c:\snort\snort.exe -W`
	
	This should produce output listing the network adapters available to Snort. The number listed under the "Interface" heading is how we will specify the desired NIC in the Snort command. Output of the snort -W command will follow the general layout:`c:\Snort\bin>snort -W`
	
	`Interface Device Description`
	`----------`
	`1 \Device\NPF_{8E90C605-F331-449F-88B3-84623E45CCA3} (VMware Virtual Ethernet A dapter)`
	`2 \Device\NPF_{085FE406-C13D-446B-A99D-85439E676520} (VMware Virtual Ethernet Ad apter)`
	`3 \Device\NPF_{C3E5FBC7-2641-4B26-9272-906CB3861AD6} (Realtek PCIe GBE Family Co ntroller) c:\Snort\bin>`
	
	Note: Your device names and descriptions will be different.

3	Run Snort to capture data involving 192.168.4.19. Type (all on one line): `snort –i <?> -l c:\snort\log –L snort.log –v -c c:\snort\etc\snort.conf host 192.168.4.19`
4	After capture, press Ctrl-C to quit Snort.

Procedure: Verifying Snort Data

Snort outputs its data into a tcpdump pcap file format, which cannot be viewed with regular text viewers. Follow these steps to ensure that Snort collected data properly.

Step	Action
1	Open Windows Explorer and browse to the log file in Windows Explorer.
2	Drag and drop the desired log file from Windows Explorer onto the Wireshark shortcut icon on your desktop.
3	Wireshark will open and should display the captured data.



Snort Alert Example

Here is a line-by-line example of a Snort alert log entry.

Line	Data
1	`[**][1:718:1] TELNET login incorrect [**]`
2	`[Classification: Attempted Information Leak] [Priority: 3]`
3	`09/16/03-03:06:08.214233 192.168.4.46:23 -> 192.168.1.102:1028`
4	`TCP TTL:64 TOS:0x0 ID:1606 IpLen:20 DgmLen:59 DF`
5	`***AP*** Seq: 0xE2057843 Ack: 0x24c4348 Win: 0x16D0 TcpLen: 20`
6	`[Xref => http://www.whitehats.com/info/IDS127]`

Snort Alert Description

Here is the same line-by-line example with a detailed explanation of the elements.

Line	Data Element	Explanation
1	1:718:1	These numbers stand for the following values: GID:SID:Rev • **GID:** Generator ID. This number corresponds to the component of Snort that identified the particular event. In this case, 1 corresponds to the main rules engine. The generator list can be found in the text file:<Snort Install Dir>/etc/generator • **SID:** Snort rule ID. This number corresponds to the specific rule used to identify the event. The number can be found in the specific rule file, located in the directory:<Snort Install Dir>/rules • **Rev:** This number identifies the revision number of the particular rule used to ID the event. The number can be found in the specific rule file, located in the directory:<Snort Install Dir>/rulesTELNET login incorrectBrief description of the event
2	Classificatio n: Attempted Information Leak	The classification of the event. The classification for an alert rule is listed in the rule file. A list can be found in: <Snort Install Dir>/etc/classification.config
	Priority: 3	The severity level of the event. A lower number indicates a higher severity level. The priority number can be defined either in the rule file itself or in classification.config.
3	09/16/03	Date of the event
	03:06:08.2142 33	Time of the event in hours, minutes, seconds and milliseconds
	192.168.4.46	Source IP address
	23	Source port number
	192.168.1.102	Destination IP address
	1028	Destination port number
4	TCP	Protocol that contained the data generating the alert, usually TCP or UDP
	TTL: 64	Time To Live: The number of network devices (Layer 3 and above) that a packet can pass through before it is dropped

Line	Data Element	Explanation
	TOS: 0x0	IP Type of service field. The possible values and their meanings follow: • 0x0: Normal service • 0x10: Minimize delay • 0x08: Maximize throughput • 0x04: Maximize reliability • 0x02: Minimize monetary cost
	ID: 1606	An identification number for the IP datagram that generated the alert. Most operating systems increment this value by 1 every time they send an IP datagram. The value is used when dealing with IP fragments.
	IpLen: 20	Length of the IP header in bytes, in this case 20.
	DgmLen: 59 DF	Length of the entire IP datagram in bytes, in this case 59
5	***AP***	This section shows any active TCP flags. A letter represents an active flag, and an asterisk represents an inactive flag. The letters and the flag they represent are: • C – CWR (Congestion Window Reduced) • E – ECN-Echo (Express Congestion Notification Echo) • U – URG (Urgent) • A – Ack (Acknowledgment) • P – PSH (Push) • R – RST (Reset) • S – SYN (Synchronize) • F – FIN (Finish)
	Seq: 0xE2057843	TCP sequence number, listed in hexadecimal format. This identifies the TCP segment and where it fits with other segments that are part of the same session.
	Ack: 0x24c4348	Acknowledgment number. The TCP/IP stack generates this number by incrementing the sequence number of the last segment received in the session by one.
	Win: 0x16D0	Window length. This number advertises the available space in the sending computer's receive buffer.
	TcpLen: 20	Length of the TCP header in bytes, in this case 20.
6	[Xref => http:// www.wh itehats. com/i nfo/IDS127]	A reference link to a Web page that gives more details on this particular alert.

Snort Options for Running Against a Capture File

Snort is run from the command line. A large number of command line options are available. Those most pertinent to this lesson are listed in this chart. Read through the Snort documentation to learn some of the other available options.

Options	Description
-c	Path to the snort.conf configuration file Usage: `–c <full path>\snort.conf`
-r	Read in a capture file Usage: `-r <filename.cap>`
-l	Specify the directory where output will be placed Usage: `-l <full path>`

Running Snort Against a Capture File

The full syntax for running Snort against a capture file is listed below, with the minimum required options.

```
snort –c <full path>\snort.conf –r <path\filename.cap>
```

Also, specify an output directory using the "-l" option. For example:

```
snort –c c:\snort\etc\snort.conf –r e:\evidence\log.cap
–l c:\snort\log
```

Custom Snort Rules

Snort rules can be added by creating a new ".rules" file with individual rules within it and adding that file to the snort.conf configuration file. The other option is to add a new line to a current rules file. In either case, the syntax for the rule itself is the same. This lesson describes the basic syntax for a simple ASCII or hexadecimal content rule. For further details, reference the Snort documentation at http://www.snort.org.

Snort Rule Syntax

Here is the general syntax for a Snort rule:

```
action proto src_ip src_port -> dst_ip dst_port (msg;
payload_detect_pattern; modifier; sid; rev;)
```

Here is an example:

```
alert ip any any -> any any (msg:“Bomb threat”;
content:“bomb”; nocase; sid:9999; rev:1;)
```

Field	Meaning
action	Action taken by the IDS when it achieves a signature match;“alert” is the most common value, which means that a log entry will be generated
proto	The lowest level protocol in which Snort will search; options include IP, TCP, UDP and ICMP
src ip	Source IP address or network address
src port	Source TCP or UDP port
dst ip	Destination IP address or network address
dst port	Destination TCP or UDP port
msg	Text message displayed in the log entry if there is a rule match
payload detect	Option for detecting specific content in a packet; it is most common to search for ASCII or hex content
modifier	An option that modifies how payload detection occurs
sid	A number that identifies the rule
rev	The revision number for the rule

Snort Rule Syntax: IP Address and Ports

IP addresses can be listed as a specific address, an address range, or a variable corresponding to a host or network address specified in the snort.conf file. Network addresses are listed in CIDR notation. Port numbers can also be listed as the number or a variable that references the configuration file.

The value of “any” is acceptable for both addresses and port numbers to indicate that Snort should not consider the address/port when determining if a packet matches the rule.

A “!” can also be placed before an IP address or port number to search for any packet that does not contain the value.

Snort Rule Syntax: Message

The alert message field (msg:<"text>";) can be arbitrarily defined but should be descriptive enough that anyone analyzing the log can quickly determine why the alert may have been generated. For example:

msg:"This is my first custom rule;"

Snort Rule Syntax: SID and Revision Numbers

The SID and revision numbers can be arbitrarily set. However, the SID should be unique among rules currently in use.

Snort Rule Syntax: Payload Detection

The primary option for payload detection is the content option. It is used as follows.

- ASCII text criteria:
  - `content:"<insert text here>";`
- ASCII text criteria example:
  - `content:"bomb";`
- Hex criteria syntax:
  - `content:"|<insert hex digits here>|";`
- Hex criteria example:
  - `content:"|5F 27 A8 89|";`

The main difference is the use of the "|" character. Hex values must be bordered by this character.

Snort Rule Syntax: Payload Detection Modifiers

Several modifiers exist for payload detection rules. These two will be of the most value to you for immediate use:

- **Rawbytes:** Snort evaluates the packet without any preprocessing. The syntax is:
  - `rawbytes;`
- **Nocase:** Snort evaluates for text content regardless of uppercase or lowercase characters. The syntax is:
  - `nocase;`

Snort Rule Syntax: Regular Expression Payload Detection

Snort can use PCRE or PERL Compatible Regular Expressions to search for packet content as well as ASCII or hex literals. This is done via the following:

- PCRE syntax:
- pcre:"/<expression>/<modifiers>";
- PCRE example:
- pcre:"/[0-9]/";

Procedure: Creating a Custom Snort Rule

Since you will use Snort through the command line, you will use the Windows command line interface to create a new custom rule and modify the snort.conf file to implement that rule by following these steps.

Step	Action
1	Open a Windows command prompt and navigate to the Snort rules directory: `cd c:\Snort\rules`
2	Use Windows Notepad to create the file "football.rules." Type: `notepad football.rules` If the file "football.rules" does not exist, Notepad will display an alert window asking if you want to create the new file. Click "Yes".
3	Type this rule as a single line with no line breaks or carriage returns: `alert ip any any -> any any (msg:"Football Alert"; content:"NFL"; nocase; sid:999999; rev:1;)`
4	Save the file and exit by pressing <ctrl-s> and then <alt-F4>

Step	Action

5	Ensure the rule file was created. Type:
	
	`dir /s football.rules`
	
	This should produce output similar to:
	
	`c:\Snort\rules>dir /s football.rules`
	`  Volume in drive C has no label.`
	`  Volume Serial Number is D023-330B`
	
	`Directory of c:\Snort\rules`
	`07/25/2013 10:44 AM  93 football.rules`
	`  1 File(s)  93 bytes`
	
	`  Total Files Listed:`
	`  1 File(s)  93 bytes`
	`  0 Dir(s)  35,282,817,024 bytes free`
	
	`c:\Snort\rules>`

6	Edit the snort.conf file to activate the new rule. Type: `write c:\snort\etc\snort.conf` Note: Use wordpad here to preserve the legibility of the snort.conf file.
7	Scroll toward the bottom of the config file that enables each rule set. Enable the new rule. Type: `include $RULE_PATH\football.rules`
8	Press Ctrl-s to save and Alt-F4 to close wordpad.

Procedure: Testing the New Custom Snort Rule

Follow these steps to test the implementation of the football rule.

Step	Action

1	At the Windows command line, determine which NIC is your out-of-band monitor. Type:
	`c:\snort\snort.exe -W`
	This should produce output listing the network adapters available to Snort. The number listed under the "Interface" heading is how we will specify the desired NIC in the Snort command. Output of the snort -W command will follow the general layout:
	`c:\Snort\bin>snort -W`
	`Interface Device Description`
	`----------`
	`1 \Device\NPF_{8E90C605-F331-449F-88B3-84623E45CCA3} (VMware Virtual Ethernet A dapter)`
	`2 \Device\NPF_{085FE406-C13D-446B-A99D-85439E676520} (VMware Virtual Ethernet Ad apter)`
	`3 \Device\NPF_{C3E5FBC7-2641-4B26-9272-906CB3861AD6} (Realtek PCIe GBE Family Co ntroller) c:\Snort\bin>`
	Note: Your device names and descriptions will be different.

2	Execute snort with the proper configuration to monitor the proper NIC using the proper configuration file that includes the new football rules you created. **Note:** Type the following all on one line. `c:\snort\snort.exe –i <?> -l c:\snort\log –L snort.log –v-c c:\snort\etc\snort.conf host <IP of interest>`
3	Using the browser of your choice, browse to pages that contain "NFL," such as nfl.com or your favorite team site.
4	When browsing is complete, terminate sort using Ctrl-C.
5	Review the alerts and captured packets. The data resides in the snort/log directory. The alert file contains the alerts. There is also a pcap formatted file that contains the complete packets that triggered the alerts. Note that this file only contains those packets that triggered the alerts; it is not a full-packet capture.

Attacks and Their Artifacts

Attack Examples

In the attack phase, an intruder takes actions necessary to gain unauthorized access to a system, elevate privileges or damage a system. This section explains the goals and methodologies an intruder uses to perform these actions.

Attack Goals

An attack is an action taken to further one of the following goals:

- **Unauthorized access:** Obtaining access to a resource (system, network, data, etc.) illegally, against policy or otherwise unauthorized by the organization/individual owning that resource
- **Access privilege:** Obtaining the ability to manipulate a resource (change, delete, deactivate, etc.) to an extent not authorized by the organization/individual owning that resource
- **Denial of service (DoS):** Preventing a resource from being available to fulfill its purpose either temporarily or permanently

Unauthorized access and privilege are often both goals of the same attack. In many instances, they can be achieved through a single attack technique. Other times, an initial attack is used to gain access, and subsequent attacks are used to obtain the proper privilege.

Terminology Note

The term denial of service (DoS) is often used to indicate an attack that is performed by flooding a network link or interface. The term distributed denial of service (DDoS) is used when this attack is sourced from many different hosts.

This text uses the term denial of service (DoS) in the general sense of any action that prevents a target from performing its normal actions.

Credential Guessing/Cracking

One of the most direct ways to access an information system is using legitimate credentials of an existing account. Obtaining the credentials is sometimes as simple as guessing at the value and attempting to authenticate. If the authentication fails, the attacker guesses again. This is sometimes called a password attack; however, the general method can be used for factors other than passwords.

Factor guessing or cracking techniques require taking the following actions:

- The attacker generates a set of values that represent possible legitimate authentication factors.
- The attacker tests those values against the authentication system or a stolen set of password hashes to determine which ones, if any, are correct.

Possible artifacts of authentication factor cracking or guessing include:

- Large numbers of failed authentication attempts for a single account, as seen in the logs of the authentication system, e.g., the Windows Security Event Log, or the /var/log/secure file on a Linux or Unix system with SSH
- Failed authentication attempts for one or more nonexistent user account names, as seen in the logs of the authentication system
- Failed authentication attempts that show a series of passwords matching a pattern indicative of an attack (such as the aa, ab, ac, etc., that might be seen in a brute force value set); authentication logs do not always record the password that was entered, but they may be visible in a network sniffer log
- IDS logs referencing a password or authentication attack
- User accounts that are locked out due to an unusually high number of failures
- A list of passwords or password hashes found in a text file in an abnormal location
- The presence of password/hash dumping utilities such as pwdump (pwdump.dll)
- Authentication attempts (successful or failed) at abnormal times or those that the authorized user of the account does not recall

Credential Injection/Modification

Almost all systems that include a mechanism for authenticating users/programs also include a mechanism for creating new accounts when necessary. If an attacker has access to this mechanism, he/she can use that mechanism to create new accounts. Techniques for accomplishing this include:

- Calling a helpdesk and requesting the creation of an account. This will usually require that the attacker masquerade as someone who can legitimately request such an action.
- Using an online mechanism to request an account. This can be done to gain initial access to information on Web sites that provide information as long as a user registers and sometimes pays. Some Web sites might require that a user validate identify for registration, using personal information. Others allow a user to input whatever information he/she chooses to provide.

- Directly creating user accounts using an available administrative utility, e.g., the Active Directory Users and Computers console, or directly inserting them into a list of users, e.g., /etc/passwd.

Possible artifacts of credential injection and modification include:

- Existence of a user account in an account repository for which there is no legitimate authorized user
- Existence of a user account in an account repository that does not match the account naming convention for the organization
- Recorded logins of an account for which there is no legitimate authorized user as seen in OS or application security/authentication logs
- Log entries referencing account creation

Data Flooding

Flooding is the act of sending as much data as possible to a target to disrupt its normal operations. The target could be a network link or one or more specific hosts. Techniques for accomplishing this include:

- Flooding a network link with packets from one or more sources to prevent legitimate traffic from being processed at normal speeds or even at all
- Flooding a system (usually a server) with as many requests for a service as possible to prevent that service from responding to legitimate requests

Possible artifacts of flooding include:

- Large amounts of network traffic as seen in a sniffer, router or firewall log; this traffic may come from one host or many hosts
- IDS alerts indicating a DoS or DDoS attack
- Errors in the log files (for the service or application that was attacked) indicating failure due to excessive requests
- Excessively large application log files

Buffer Overflow Attack

A buffer is a temporary storage area, usually in RAM, allocated for the manipulation of data within a process. For instance, when logging into an e-mail server, there might be a 32-character space for a username. This limitation established in the code of the e-mail server application controls how much input will be accepted from the user.

A buffer overflow attack purposely sends an entry too large for the buffer to hold. It sends it so that a portion of the entry is written to the target computer where program instruction code is stored.

An attacker uses this method to cause the execution of his code intentionally. The result of this code execution could be anything, but it will often be a DoS, a command terminal session sent back to the attacker's computer, or the injection of a dynamic link library (DLL) or other program code into the remote process.

Possible artifacts of buffer overflow attacks include:

- Unexplained errors in the log files for the application or service that was attacked; if an OS service was attacked, log entries may exist in the OS's main log files, in supplementary crash logs such as core dumps in a Linux environment, or in Dr. Watson logs in a Windows environment
- IDS alerts indicating a buffer overflow, shellcode or "no operation" (NOOP)
- Sniffer logs that show large blocks of repetitive data, such as 0x90 (the most common NOOP instruction in hex value) or other hex values
- Sniffer logs that show blocks of data that do not conform to normal rules for the network protocol being used
- IDS alerts or sniffer logs showing common post-attack events such as reverse shells, DLL transfer or OS commands

Attacker Communication

By characterization, malware is aimed to perform malicious actions on an infected host. In most cases, these actions need to be done without alerting the user. In order to remain undetected, communication between an attacker and the malware needs to be done in a manner that is as stealthy as possible. As a result, malware authors increasingly use covert communication channels in order to hide their communication from infected users and network analysts. The goal of the Command & Control (C2) communication channel is to send situational awareness information back to the attacker and to allow the attacker to send commands forward to the client.

Attacker methods to communicate:

- **More overt:** direct TCP and UDP sockets, typically over ports often open in a firewall (80,443) with either clear-text payloads or encoded/encrypted payloads
- **Less overt:** use covert channels to hide information locally on a system or mask traffic as it passes over a network (unused header fields, unused fields in existing packets, steganography techniques, DNS tunneling)

- **Steganography:** used to hide message inside digital images or other media files
- **DNS tunneling:** the ability to encode the data of other programs or protocols in DNS queries and response

Attacker C2 Detection

C2 detection falls broadly into two categories: signature-based and non-signature based. In signature-based detection methods, malware C2 is detected by looking for known patterns of behavior, or "Signatures". Signatures are generated from known malware samples, and then new traffic is compared to these signatures. In order to effectively work IDS sensors must be configured to spot protocol anomalies. Non-signature based algorithms instead look for anomalies compared to normal behavior. The main disadvantage of using a signature based detection method is that these detection systems are usually not very effective at detecting new, or updated, malware. Every time a new piece of malware is discovered, or an existing piece updates itself, the signatures have to be recreated. If the new variant is not discovered, then it is unlikely to be detected by these systems. This is where non-signature based detection comes in. In these systems, the algorithms look for behavior that is not expected, rather than looking for particular known behavior, or looking for a specific type of behavior without the use of signatures.

List of a few security measures an organization should implement to detect and defend against C2 exploitation.

Security Measures for C2 Detection and Defense

- Configure IDS sensors to spot protocol anomalies
  - i.e IRC chat over port 80
- Use netflow to profile your organizations network environment to spot traffic anomalies
- Configure network so all outbound traffic passes through a "choke point"
- Analyze large data sets over extended periods
  - Log everything from all machines and devices
  - Collect full packet network traffic at boundaries
  - Maintain logs for an extended period of time
  - Store network traffic alerts in logs analytic systems for analysis and inspection
- Deploying data loss prevention (DLP) tools at the network perimeter, to identify sensitive data leaving the network.