2.5.2 Network Monitoring Facts

The goal of monitoring is to keep track of conditions on the network, identify situations that might signal potential problems, pinpoint the source of problems, and locate areas of your network that might need to be upgraded or modified. As you monitor your network, look for your top talkers and listeners. *Top talkers* are computers that send the most data, either from your network or into your network. *Top listeners* are hosts that are receiving most of the data by streaming or downloading large amounts of data from the internet. It is important to know which computers are the big receivers and senders of information because it is a good way to tell if something is wrong on your network. An unauthorized system that is sending large amounts of data to locations outside of your network could be a sign of a data breach.

The following table lists some tools you can use to check the health of your network:

Tool	Description
Logs	*Logs* are a record of events that have occurred on a system. Logging capabilities are built into operating systems, services, and applications. Log entries are generated in response to changes in configuration, system state, or network conditions. - By default, some logging is enabled and performed automatically. To gather additional information, you can usually enable more extensive logging. - Many systems have logs for different purposes, such as a system log for operating system entries, a security log for security-related entries, and an application log (also called a performance log) for events related to specific services and processes, such as connections from a web server. - Logging requires system resources (processor, memory, and disk). You should only enable additional logging based on information you want to gather, and you should disable logging after you obtain the information you need. - Logs must be analyzed to be useful; only by looking at the logs will you be able to discover problems. Depending on the log type, additional tools might be available to analyze logs for patterns. - Logs should show a narrow view of information. If a log contains too much information, then issues could be lost in all of the information and harder to spot. - The IETF syslog standard is a standard for managing and sending log messages from one computer system to another. Software can analyze syslog messages and notify administrators of problems or performance.
Load Tester	A *load tester* simulates a load on a server or service. For example, the load tester might simulate a large number of client connections to a website, simulate test file downloads for an FTP site, or simulate large volumes of email. Use a load tester to make sure that a system has sufficient capacity for expected loads. A load tester can even estimate failure points where the load is more than the system can handle.
Throughput Tester	A *throughput tester* measures the amount of data that can be transferred through a network or processed by a device (such as the amount of data that can be retrieved from a disk in a specific period of time). On a network, a throughput tester sends a specific amount of data through the network and measures the time it takes to transfer that data. This creates a measurement of the actual bandwidth of the network. Use a throughput tester to validate the bandwidth on your network and identify when the bandwidth is significantly below what it should be. A throughput tester can help you identify when a network is slow but will not give you sufficient information to identify why it is slow.
Packet Sniffer	A *packet sniffer* is software that captures or records frames that are transmitted on the network. Use a packet sniffer to: - Identify the types of traffic on a network. - View the exchange of packets between communicating devices. For example, you can capture frames related to DNS and view the exact exchange of packets for a specific name resolution request. - Analyze packets sent to and from a specific device. - View packet contents. A packet sniffer is typically run on one device with the intent of capturing frames for all other devices on a subnet. Using a packet sniffer in this way requires the following configuration changes: - The NIC must be configured in promiscuous mode. By default, a NIC will only accept frames addressed to itself. Normally, packet sniffer software will configure the NIC in promiscuous mode (sometimes called p-mode). In p-mode, the NIC will process every frame it sees. - The switch must be configured with *port mirroring*. When using a switch, the switch will forward packets only to the switch port that holds a destination device. When your packet sniffer is connected to a switch port, it will not see traffic sent to other switch ports. To configure the switch to send all frames to the packet sniffing device, configure *port mirroring* on the switch; all frames sent to all other switch ports will be forwarded on the mirrored port. A hub will forward packets received from any port to all other ports. If the packet sniffer is connected to a hub, it will already see all frames sent to any device on the hub.
Protocol Analyzer	A *protocol analyzer* is a special type of packet sniffer that captures transmitted frames. A protocol analyzer is a passive device in that it copies frames and allows you to view frame contents but does not allow you to modify and retransmit frames (activities that are used to perform an attack). Use a protocol analyzer to: - Check for specific protocols on the network, such as SMTP, DNS, POP3, and ICMP. - Find devices that might be using restricted protocols (such as ICMP) or legacy protocols (for example, IPX/SPX or NetBIOS). - Analyze traffic that might be sent by attackers.

	- Identify frames that might cause errors. - Determine which flags are set in a TCP handshake. - Detect many malformed or fragmented packets. - Examine the data contained within a packet. - Identify users that are connecting to unauthorized websites. - Discover cleartext passwords allowed by protocols or services. - Identify unencrypted traffic that includes sensitive data. - Troubleshoot communication problems or investigate the source of heavy network traffic. A protocol analyzer shows the traffic that exists on the network and the source and destination of that traffic. It does not tell you if the destination ports on a device are open unless you see traffic originating from that port. For example, seeing traffic addressed to port 80 of a device does not automatically mean the firewall on that device is open or that the device is responding to traffic directed to that port. When using a protocol analyzer, you can filter the frames so that you see only the frames with information of interest. - Filters can be configured to show only frames or packets to or from specific addresses or frames that include specific protocol types. - A *capture* filter captures only the frames identified by the filter. Frames that do no match the filter criteria will not be captured. - A *display* filter shows only the frames that match the filter criteria. Frames that do not match the filter criteria are still captured, but are not shown. - The results of a capture can be saved in order to analyze frames at a later time or on a different device.
Command Line Tools	There are several command line tools that can help you determine the condition of your network. Depending on the operating system, these command may vary in their format, but operate in a similar manner. - The **ping** (Packet INternet Groper) command can be used to verify network connectivity between two hosts or nodes. It can also be used to test network latency. - The **netstat** (NETwork STATistics) command displays statistical information describing TCP network connections, routing tables, network interfaces and network protocols. - This utility is mostly obsolete in Linux, but still included in many distributions - In Linux, netstat has been superceded by the **ss** and **ip** commands. - The **tracert** (TRACE RouTe) command displays the IP route to a destination host or node. - In Linux and Mac OS, the command is **traceroute**. - The **nslookup** (Name Server LOOKUP) command will query a DNS to obtain the IP address for a given domain name, or to obtain a domain name for a given IP address. - In Linux, the **dig** command gives similar information. - The **arp** (Address Resoultion Protocol) command is used to display and modify the ARP table entries on the local host. The ARP table maps internet IP addresses with physical MAC addresses. - The **ipconfig** (Internet Protocol CONFIGuration) command is used to display a host's current TCP/IP configuration values and to refresh DHCP and DNS settings. - In Linux and Mac OS, the command is **ifconfig**. However, the newer **ip** command has more features and will eventually replace ifconfig. - The **tcpdump** utility is a network sniffer and analyzer. It displays a description of packet contents on a network interface. - The **tcpdump** utility is packaged in most Linux and Mac OS distributions. - **tcpdump** is not part of the Windows OS but can be readily downloaded from the internet. - The **nmap** (Network MAPper) utility is a network security scanner. It is commonly used to scan a system to determine which TCP ports are open. - The **nmap** utility is packaged in most Linux distributions. - For Windows and MacOS, the nmap utility can be readily downloaded from the Internet. - The **netcat** utility can read and write data across both TCP and UDP network connections. - The netcat utility is not native to any operating systems; you can download it from the internet.