- Token Ring, IEEE 802.5, is an older LAN implementation that uses a token-passing technology.
- FDDI is a LAN and MAN technology, usually used for backbones, that uses token-passing technology and has redundant rings in case the primary ring goes down.
- TCP/IP is a suite of protocols that is the de facto standard for transmitting data across the Internet. TCP is a reliable, connection-oriented protocol, while IP is an unreliable, connectionless protocol.
- Data is encapsulated as it travels down the network stack on the source computer, and the process is reversed on the destination computer. During encapsulation, each layer adds its own information so the corresponding layer on the destination computer knows how to process the data.
- Two main protocols at the transport layer are TCP and UDP.
- UDP is a connectionless protocol that does not send or receive acknowledgments when a datagram is received. It does not ensure data arrives at its destination. It provides "best-effort" delivery.
- TCP is a connection-oriented protocol that sends and receives acknowledgments. It ensures data arrives at the destination.
- ARP translates the IP address into a MAC address (physical Ethernet address), while RARP translates a MAC address into an IP address.
- ICMP works at the network layer and informs hosts, routers, and devices of network or computer problems. It is the major component of the ping utility.
- DNS resolves hostnames into IP addresses and has distributed databases all over the Internet to provide name resolution.
- Altering an ARP table so an IP address is mapped to a different MAC address is called ARP poisoning and can redirect traffic to an attacker's computer or an unattended system.
- Routers link two or more network segments, where each segment can function as an independent network. A router works at the network layer, works with IP addresses, and has more network knowledge than bridges, switches, or repeaters.
- IPv4 uses 32 bits for its addresses, whereas IPv6 uses 128 bits; thus, IPv6 provides more possible addresses with which to work.
- NAT is used when organizations do not want systems to know internal hosts' addresses, and it enables organizations to use private, nonroutable IP addresses.
- Subnetting allows large IP address ranges to be divided into smaller, logical, and easier-to-maintain network segments.
- Dedicated links are usually the most expensive type of WAN connectivity method because the fee is based on the distance between the two destinations rather than on the amount of bandwidth used. T1 and T3 are examples of dedicated links.
- Frame relay and X.25 are packet-switched WAN technologies that use virtual circuits instead of dedicated ones.

- ATM transfers data in fixed cells, is a WAN technology, and transmits data at very high rates. It supports voice, data, and video applications.
- Circuit-switching technologies set up a circuit that will be used during a data transmission session. Packet-switching technologies do not set up circuits—instead, packets can travel along many different routes to arrive at the same destination.
- Three main types of multiplexing are statistical time division, frequency division, and wave division.

Questions

Please remember that these questions are formatted and asked in a certain way for a reason. Keep in mind that the CISSP exam is asking questions at a conceptual level. Questions may not always have the perfect answer, and the candidate is advised against always looking for the perfect answer. Instead, the candidate should look for the best answer in the list.

1. Which of the following protocols is considered connection-oriented?

A. IP

B. ICMP

C. UDP

D. TCP

2. Which of the following shows the layer sequence as layers 2, 5, 7, 4, and 3?

A. Data link, session, application, transport, and network

B. Data link, transport, application, session, and network

C. Network, session, application, network, and transport

D. Network, transport, application, session, and presentation

3. Metro Ethernet is a MAN protocol that can work in network infrastructures made up of access, aggregation, metro, and core layers. Which of the following best describes these network infrastructure layers?

A. The access layer connects the customer's equipment to a service provider's aggregation network. Aggregation occurs on a core network. The metro layer is the metropolitan area network. The core connects different metro networks.

B. The access layer connects the customer's equipment to a service provider's core network. Aggregation occurs on a distribution network at the core. The metro layer is the metropolitan area network.

C. The access layer connects the customer's equipment to a service provider's aggregation network. Aggregation occurs on a distribution network. The metro layer is the metropolitan area network. The core connects different access layers.

D. The access layer connects the customer's equipment to a service provider's aggregation network. Aggregation occurs on a distribution network. The metro layer is the metropolitan area network. The core connects different metro networks.

4. Systems that are built on the OSI model are considered open systems. What does this mean?

A. They do not have authentication mechanisms configured by default.

B. They have interoperability issues.

C. They are built with internationally accepted protocols and standards so they can easily communicate with other systems.

D. They are built with international protocols and standards so they can choose what types of systems they will communicate with.

5. Which of the following protocols work in the following layers: application, data link, network, and transport?

A. FTP, ARP, TCP, and UDP

B. FTP, ICMP, IP, and UDP

C. TFTP, ARP, IP, and UDP

D. TFTP, RARP, IP, and ICMP

6. What takes place at the data link layer?

A. End-to-end connection

B. Dialog control

C. Framing

D. Data syntax

7. What takes place at the session layer?

A. Dialog control

B. Routing

C. Packet sequencing

D. Addressing

8. Which best describes the IP protocol?

A. A connectionless protocol that deals with dialog establishment, maintenance, and destruction

B. A connectionless protocol that deals with the addressing and routing of packets

C. A connection-oriented protocol that deals with the addressing and routing of packets

D. A connection-oriented protocol that deals with sequencing, error detection, and flow control

9. Which of the following is not one of the messages exchanged during the DHCP lease process?
    i. Discover
    ii. Offer
    iii. Request
    iv. Acknowledgment
    A. All of them are exchanged
    B. None of them are exchanged
    C. i, ii
    D. ii, iii
10. An effective method to shield networks from unauthenticated DHCP clients is through the use of __________ on network switches.
    A. DHCP snooping
    B. DHCP protection
    C. DHCP shielding
    D. DHCP caching

Answers

1. **D.** TCP is the only connection-oriented protocol listed. A connection-oriented protocol provides reliable connectivity and data transmission, while a connectionless protocol provides unreliable connections and does not promise or ensure data transmission.
2. **A.** The OSI model is made up of seven layers: application (layer 7), presentation (layer 6), session (layer 5), transport (layer 4), network (layer 3), data link (layer 2), and physical (layer 1).
3. **D.** The access layer connects the customer's equipment to a service provider's aggregation network. Aggregation occurs on a distribution network. The metro layer is the metropolitan area network. The core connects different metro networks.
4. **C.** An open system is a system that has been developed based on standardized protocols and interfaces. Following these standards allows the systems to interoperate more effectively with other systems that follow the same standards.
5. **C.** Different protocols have different functionalities. The OSI model is an attempt to describe conceptually where these different functionalities take place in a networking stack. The model attempts to draw boxes around reality to help people better understand the stack. Each layer has a specific functionality and has several different protocols that can live at that layer and carry out that specific functionality. These listed protocols work at these associated layers: TFTP (application), ARP (data link), IP (network), and UDP (transport).

6. **C.** The data link layer, in most cases, is the only layer that understands the environment in which the system is working, whether it be Ethernet, Token Ring, wireless, or a connection to a WAN link. This layer adds the necessary headers and trailers to the frame. Other systems on the same type of network using the same technology understand only the specific header and trailer format used in their data link technology.
7. **A.** The session layer is responsible for controlling how applications communicate, not how computers communicate. Not all applications use protocols that work at the session layer, so this layer is not always used in networking functions. A session layer protocol sets up the connection to the other application logically and controls the dialog going back and forth. Session layer protocols allow applications to keep track of the dialog.
8. **B.** The IP protocol is connectionless and works at the network layer. It adds source and destination addresses to a packet as it goes through its data encapsulation process. IP can also make routing decisions based on the destination address.
9. **B.** The four-step DHCP lease process is
    1. **DHCPDISCOVER message:** This message is used to request an IP address lease from a DHCP server.
    2. **DHCPOFFER message:** This message is a response to a DHCPDISCOVER message, and is sent by one or numerous DHCP servers.
    3. **DHCPREQUEST message:** The client sends this message to the initial DHCP server that responded to its request.
    4. **DHCPACK message:** This message is sent by the DHCP server to the DHCP client and is the process whereby the DHCP server assigns the IP address lease to the DHCP client.
10. **A.** DHCP snooping ensures that DHCP servers can assign IP addresses to only selected systems, identified by their MAC addresses. Also, advance network switches now have the capability to direct clients toward legitimate DHCP servers to get IP addresses and to restrict rogue systems from becoming DHCP servers on the network.

CHAPTER 12

Wireless Networking

This chapter presents the following:

- Wireless networking
- Wireless LAN security
- Cellular networks
- Satellite communications

> *When wireless is perfectly applied the whole earth will be converted into a huge brain…*
>
> —Nikola Tesla

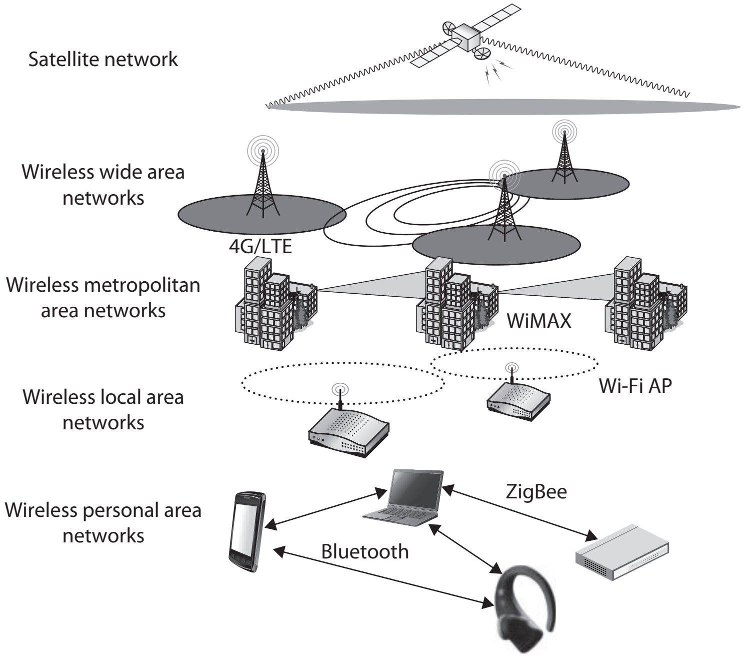
Wireless communications take place much more often than most people realize, and they involve a vast number of technologies working over a multitude of radio frequency ranges. These radio signals occupy frequency bands that may be shared with microwave, satellite, radar, and ham radio use, for example. We use these technologies for satellite communications, cellular phones, metropolitan and local area networking, and even for locking doors and controlling lights in our smart homes, as illustrated in Figure 12-1. All these interconnected networks rely on different communications techniques, using different radio frequencies and implementing different protocols. This extremely complex ecosystem makes many of our modern conveniences possible, but also introduces significant security challenges.

In this chapter, we'll cover the fundamentals of radio communications techniques and the most important protocols you should be aware of. We then put this theoretical information into real-world contexts as we discuss the opportunities and challenges they represent. Along the way, we'll talk about security threats and how to mitigate them.

Wireless Communications Techniques

Wireless communication involves transmitting information via radio waves that move through free space. These radio signals are typically described in terms of frequency and amplitude. The *frequency* of a signal indicates how many radio waves travel through a fixed place each second (i.e., how close each radio wave is to the one before it). Frequency is measured in hertz (Hz) and dictates the amount of data that can be carried and how far. The higher the frequency, the more data the signal can carry, but the shorter its range.

Figure 12-1
Various wireless networks


Satellite network
Wireless wide area networks
4G/LTE
Wireless metropolitan area networks
WiMAX
Wireless local area networks
Wi-Fi AP
Wireless personal area networks
ZigBee
Bluetooth

The *amplitude* of a radio signal indicates its power, which in turn dictates how far it can go. Amplitude usually is measured in watts or milliwatts (one-thousandth of a watt), but you may also see it expressed in decibels per milliwatt (dBm or dBmW), which is a measure of comparison to one milliwatt. For example, a wireless access point may allow you to configure the transmit power in increments from 0 dBm (1 mW) up to 23 dBm (200 mW).

In a wired network, each computer and device has its own cable connecting it to the network in some fashion. In wireless technologies, each device must instead share the allotted radio frequency spectrum with all other wireless devices that need to communicate. This spectrum of frequencies is finite in nature, which means it cannot grow if more and more devices need to use it. The same thing happens with Ethernet—all the computers on a segment share the same medium, and only one computer can send data at any given time. Otherwise, a collision can take place. Wired networks using Ethernet employ the CSMA technology (described in Chapter 11). Wireless LAN (WLAN) technology is actually very similar to Ethernet, but it uses CSMA/CA (collision avoidance). The wireless device sends out a broadcast indicating it is going to transmit data. This is received by other devices on the shared medium, which causes them to hold off on transmitting information. It is all about trying to eliminate or reduce collisions.

A number of techniques have been developed to allow wireless devices to access and share this limited amount of medium for communication purposes. The goal of each of these wireless technologies is to split the available frequency into usable portions, since it is a limited resource, and to allow the devices to share those portions efficiently. The most popular approach is called spread spectrum, though orthogonal frequency division multiplexing (OFDM) is widely used also. Because spread-spectrum technology is so prevalent, we'll get into it in a fair amount of detail in the next section.

Spread Spectrum

Radio frequencies cover a wide range, or *spectrum*, of frequencies. Some parts of this spectrum are allocated by national governments or international agreements for specific purposes. A radio *frequency band* is a subset of the radio spectrum designated for a specific use. For example, the range of radio frequencies between 1.8 and 29.7 megahertz (MHz) is almost universally considered the amateur radio band. A well-known frequency band is frequently labeled using just a single frequency, such as when we refer to the 2.4-GHz band used in many Wi-Fi systems. This band actually corresponds to the range of frequencies between 2.4 and 2.5 GHz. The challenge is how to dynamically allocate individual frequencies to specific sets of transmitters and receivers without them stepping all over each other. This is where spread-spectrum techniques come in handy.

Spread spectrum means that something is distributing individual signals across the allocated frequencies in some fashion. So, when a spread-spectrum technique is used, the sender spreads its data across the frequencies over which it has permission to communicate. This allows for more effective use of the available spectrum, because the sending system can use more than one frequency at a time.

Think of spread spectrum in terms of investments. In conventional radio transmissions, all the data is transmitted on a specific frequency (as in amplitude modulated [AM] radio systems) or on a narrow band of frequencies (as in frequency modulated [FM] radio). This is like investing only in one stock; it is simple and efficient, but may not be ideal in risky environments. The alternative is to diversify your portfolio, which is normally done by investing a bit of your money in each of many stocks across a wide set of industries. This is more complex and inefficient, but can save your bottom line when the stock in one of your selected companies takes a nose-dive. This example is akin to direct sequence spread spectrum (DSSS), which we discuss in an upcoming section.

There is in theory another way to minimize your exposure to volatile markets. Suppose the cost of buying and selling was negligible. You could then invest all your money in a single stock, but only for a brief period of time, sell it as soon as you turn a profit, and then reinvest all your proceeds in another stock. By jumping around the market, your exposure to the problems of any one company are minimized. This approach would be comparable to frequency hopping spread spectrum (FHSS), discussed next. The point is that spread-spectrum communications are used primarily to reduce the effects of adverse conditions such as crowded radio bands, interference, and eavesdropping.

Frequency Hopping Spread Spectrum

Frequency hopping spread spectrum (FHSS) takes the total amount of spectrum and splits it into smaller subchannels. The sender and receiver work at one of these subchannels for a specific amount of time and then move to another subchannel. The sender puts the first piece of data on one frequency, the second on a different frequency, and so on. The FHSS algorithm determines the individual frequencies that will be used and in what order, and this is referred to as the sender and receiver's *hop sequence*.

Interference is a large issue in wireless transmissions because it can corrupt signals as they travel. Interference can be caused by other devices working in the same frequency space. The devices' signals step on each other's toes and distort the data being sent. The FHSS approach to this is to hop between different frequencies so that if another device is operating at the same frequency, it will not be drastically affected. Consider another analogy: Suppose George and Marge work in the same room. They could get into each other's way and affect each other's work. But if they periodically change rooms, the probability of them interfering with each other is reduced.

A hopping approach also makes it much more difficult for eavesdroppers to listen in on and reconstruct the data being transmitted when used in technologies other than WLAN. FHSS has been used extensively in military wireless communications devices because the only way the enemy could intercept and capture the transmission is by knowing the hopping sequence. The receiver has to know the sequence to be able to obtain the data. But in today's WLAN devices, the hopping sequence is known and does not provide any security.

So how does this FHSS stuff work? The sender and receiver hop from one frequency to another based on a predefined hop sequence. Several pairs of senders and receivers can move their data over the same set of frequencies because they are all using different hop sequences. Let's say you and Marge share a hop sequence of 1, 5, 3, 2, 4, and Nicole and Ed have a sequence of 4, 2, 5, 1, 3. Marge sends her first message on frequency 1, and Nicole sends her first message on frequency 4 at the same time. Marge's next piece of data is sent on frequency 5, the next on 3, and so on until each reaches its destination, which is your wireless device. So your device listens on frequency 1 for a half-second, and then listens on frequency 5, and so on, until it receives all of the pieces of data that are on the line on those frequencies at that time. Ed's device is listening to the same frequencies but at different times and in a different order, so his device never receives Marge's message because it is out of sync with his predefined sequence. Without knowing the right code, Ed treats Marge's messages as background noise and does not process them.

Direct Sequence Spread Spectrum

Direct sequence spread spectrum (DSSS) takes a different approach by applying sub-bits to a message. The sub-bits are used by the sending system to generate a different format of the data before the data is transmitted. The receiving end uses these sub-bits to reassemble the signal into the original data format. The sub-bits are called *chips*, and the sequence of how the sub-bits are applied is referred to as the *chipping code*.

When the sender's data is combined with the chip, the signal appears as random noise to anyone who does not know the chipping sequence. This is why the sequence is sometimes called a pseudo-noise sequence. Once the sender combines the data with the chipping sequence, the new form of the information is modulated with a radio carrier signal, and it is shifted to the necessary frequency and transmitted. What the heck does that mean? When using wireless transmissions, the data is actually moving over radio signals that work in specific frequencies. Any data to be moved in this fashion must have a carrier signal, and this carrier signal works in its own specific range, which is a frequency. So you can think of it this way: once the data is combined with the chipping code, it is put into a car (carrier signal), and the car travels down its specific road (frequency) to get to its destination.

Spread Spectrum Types

This technology transmits data by "spreading" it over a broad range of frequencies:

- FHSS moves data by changing frequencies.
- DSSS takes a different approach by applying sub-bits to a message and uses all of the available frequencies at the same time.

The receiver basically reverses the process, first by demodulating the data from the carrier signal (removing it from the car). The receiver must know the correct chipping sequence to change the received data into its original format. This means the sender and receiver must be properly synchronized.

The sub-bits provide error-recovery instructions, just as parity does in RAID technologies. If a signal is corrupted using FHSS, it must be re-sent; but by using DSSS, even if the message is somewhat distorted, the signal can still be regenerated because it can be rebuilt from the chipping code bits. The use of this code allows for prevention of interference, allows for tracking of multiple transmissions, and provides a level of error correction.

FHSS vs. DSSS

FHSS uses only a portion of the total spectrum available at any one time, while the DSSS technology uses all of the available spectrum continuously. DSSS spreads the signals over a wider frequency band, whereas FHSS uses a narrowband carrier that changes frequently across a wide band.

Since DSSS sends data across all frequencies at once, it has higher data rates than FHSS. The first wireless WAN standard, 802.11, used FHSS, but as data requirements increased, DSSS was implemented. By using FHSS, the 802.11 standard can provide a data throughput of only 1 to 2 Mbps. By using DSSS instead, 802.11b provides a data throughput of up to 11 Mbps.

Orthogonal Frequency Division Multiplexing

Besides spread-spectrum techniques, another common approach to trying to move even more data over wireless frequency signals is called *orthogonal frequency division multiplexing (OFDM)*. OFDM is a digital multicarrier modulation scheme that compacts multiple modulated carriers tightly together, reducing the required spectrum. The modulated signals are orthogonal (perpendicular) and do not interfere with each other. OFDM uses a composite of narrow channel bands to enhance its performance in high-frequency bands. OFDM is officially a multiplexing technology and not a spread-spectrum technology, but is used in a similar manner.

A large number of closely spaced orthogonal subcarrier signals are used, and the data is divided into several parallel data streams or channels, one for each subcarrier. Channel equalization is simplified because OFDM uses many slowly modulated narrowband signals rather than one rapidly modulated wideband signal.

OFDM is used for several wideband digital communication types such as digital television, audio broadcasting, DSL broadband Internet access, wireless networks, and 4G/5G mobile communications.

Wireless Networking Fundamentals

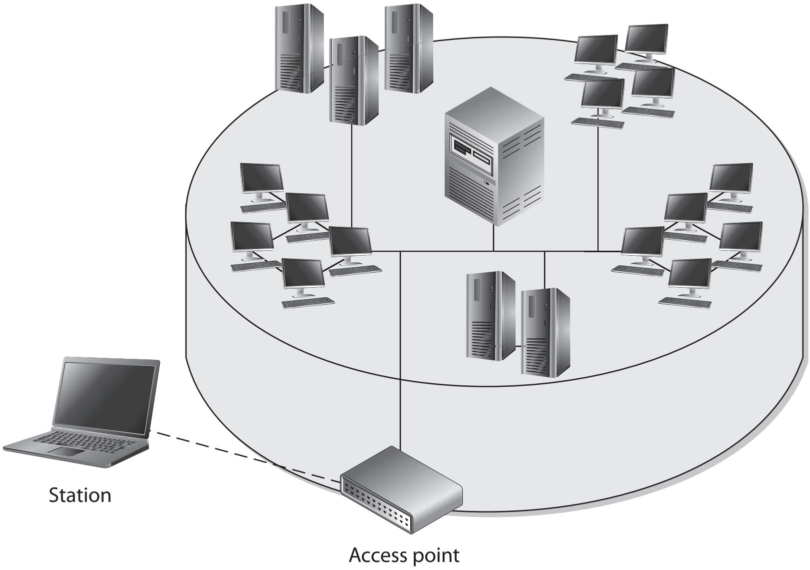
The techniques we've covered so far deal with how we create radio links between devices, but how do we build on those links to create networks? Fundamentally, there are three topologies used to build wireless networks: star, mesh, and point to point. The star topology is by far the most prevalent because it is used in both WLANs and cellular networks, both of which have endpoints connecting to a specialized network device that handles layer 2 forwarding and, in some cases, layer 3 routing. The mesh topology is common for low-power devices in close proximity to each other, such as those used in smart homes, as well as in devices that span a large area, such as environmental sensors in wildlife refuges. Finally, point-to-point wireless topologies are common when connecting buildings as part of a metropolitan area network (MAN).

Before we get into the myriad of wireless protocols that enable the various types of wireless networks, let's take a closer look at what makes a typical WLAN work.

WLAN Components

A WLAN uses a transceiver, called an *access point (AP)*, also known as a wireless access point (WAP), which connects to an Ethernet cable that is the link wireless devices use to access resources on the wired network, as shown in Figure 12-2. When the AP is connected to the LAN Ethernet by a wired cable, it is the component that connects the wired

Figure 12-2 Access points allow wireless devices to participate in wired LANs.



and the wireless worlds. The APs are in fixed locations throughout a network and work as communication beacons. Let's say a wireless user has a device with a wireless network interface card (NIC), which modulates the user's data onto radio frequency signals that are accepted and processed by the AP. The signals transmitted from the AP are received by the wireless NIC and converted into a digital format, which the device can understand.

When APs are used to connect wireless and wired networks, this is referred to as an *infrastructure WLAN*, which is used to extend an existing wired network. When there is just one AP and it is not connected to a wired network, it is considered to be in *standalone* mode and just acts as a wireless hub. An *ad hoc WLAN* has no APs; the wireless devices communicate with each other through their wireless NICs instead of going through a centralized device.

EXAM TIP Ad hoc WLANs are inherently less secure than infrastructure WLANs.

For a wireless device and AP to communicate, they must be configured to communicate over the same channel. A *channel* is a certain frequency within a given frequency band. The AP is configured to transmit over a specific channel, and the wireless device "tunes" itself to be able to communicate over this same frequency.

Any hosts that wish to participate within a particular WLAN must be configured with the proper *Service Set ID (SSID)*. Various hosts can be segmented into different WLANs by using different SSIDs. The reasons to segment a WLAN into portions are the same reasons wired systems are segmented on a network: the users require access to different resources, have different business functions, or have different levels of trust.

NOTE When wireless devices work in infrastructure mode, the AP and wireless clients form a group referred to as a Basic Service Set (BSS). This group is assigned a name, which is the SSID value.

When WLAN technologies first came out, authentication was simple and largely ineffective against many attackers. As wireless communication increased in use and many deficiencies were identified in these networks, a steady stream of improved approaches were developed and standardized. These covered both performance and security issues.

WLAN Standards

Standards are developed so that many different vendors can create various products that will work together seamlessly. Standards are usually developed on a consensus basis among the different vendors in a specific industry. The IEEE develops standards for a wide range of technologies—wireless being one of them.

The first WLAN standard, 802.11, was developed in 1997 and provided a 1- to 2-Mbps transfer rate. It worked in the 2.4-GHz frequency band, which is one of the free industrial, scientific, and medical (ISM) bands established by the International

Table 12-1 Generational Wi-Fi

Technology Supported	Wi-Fi Generation
802.11b	Wi-Fi 1
802.11a	Wi-Fi 2
802.11g	Wi-Fi 3
802.11n	Wi-Fi 4
802.11ac	Wi-Fi 5
802.11ax	Wi-Fi 6

Telecommunication Union (ITU). This means that organizations and users in most countries do not need a license to use this range. The 802.11 standard outlines how wireless clients and APs communicate; lays out the specifications of their interfaces; dictates how signal transmission should take place; and describes how authentication, association, and security should be implemented.

Now just because life is unfair, a long list of standards actually fall under the 802.11 main standard. You have probably seen the alphabet soup of 802.11a, 802.11b, 802.11g, 802.11n, 802.11ac, and 802.11ax (and a bunch of others). While the original 802.11 standard created the world of WLANs, the unrelenting pace of progress required changes and improvements over time. To try and make sense of things, the Wi-Fi Alliance created a scheme for numbering the generations of 802.11 protocols in 2018. This was done to help consumers differentiate products based on the most advanced 802.11-based technology supported by a given device. Table 12-1 lists the six generations of Wi-Fi, which we describe in the following sections.

NOTE Wi-Fi generations 1–3 are not formally defined by the Wi-Fi Alliance but are commonly understood to map to the technologies shown in Table 12-1.

802.11b

This standard was the first extension to the 802.11 WLAN standard. (Although 802.11a was conceived and approved first, it was not released first because of the technical complexity involved with this proposal.) 802.11b provides a transfer rate of up to 11 Mbps and works in the 2.4-GHz frequency range. It uses DSSS and is backward-compatible with 802.11 implementations.

802.11a

This standard uses a different method of modulating data onto the necessary radio carrier signals. Whereas 802.11b uses DSSS, 802.11a uses OFDM and works in the 5-GHz frequency band. Because of these differences, 802.11a is not backward-compatible with 802.11b or 802.11. Several vendors have developed products that can work with both 802.11a and 802.11b implementations; the devices must be properly configured or be able to sense the technology already being used and configure themselves appropriately.

As previously discussed, OFDM is a modulation scheme that splits a signal over several narrowband channels. The channels are then modulated and sent over specific frequencies. Because the data is divided across these different channels, any interference from the environment will degrade only a small portion of the signal. This allows for greater throughput. Like FHSS and DSSS, OFDM is a physical layer specification. It can be used to transmit high-definition digital audio and video broadcasting as well as WLAN traffic.

This technology offers advantages in two areas: speed and frequency. 802.11a provides up to 54 Mbps, and it does not work in the already very crowded 2.4-GHz spectrum. The 2.4-GHz frequency band is referred to as a "dirty" frequency because several devices already work there—microwaves, cordless phones, baby monitors, and so on. In many situations, this means that contention for access and use of this frequency can cause loss of data or inadequate service. But because 802.11a works at a higher frequency, it does not provide the same range as the 802.11b and 802.11g standards. The maximum speed for 802.11a is attained at short distances from the AP, up to 25 feet.

802.11g

The 802.11g standard provides for higher data transfer rates—up to 54 Mbps. This is basically a speed extension for 802.11b products. If a product meets the specifications of 802.11b, its data transfer rates are up to 11 Mbps, and if a product is based on 802.11g, that new product can be backward-compatible with older equipment but work at a much higher transfer rate.

802.11n (Wi-Fi 4)

This standard is designed to be much faster than 802.11g, with throughput at 100 Mbps, and it works at the same frequency range as 802.11a (5 GHz). The intent is to maintain some backward-compatibility with current Wi-Fi standards, while combining a mix of the current technologies. This standard uses a concept called multiple input, multiple output (MIMO) to increase the throughput. This requires the use of two receive and two transmit antennas to broadcast in parallel using a 20-MHz channel.

802.11ac (Wi-Fi 5)

The IEEE 802.11ac WLAN standard is an extension of 802.11n. It also operates on the 5-GHz band, but increases throughput to 1.3 Gbps. 802.11ac is backward compatible with 802.11a, 802.11b, 802.11g, and 802.11n, but if in compatibility mode it slows down to the speed of the slower standard. A major improvement is the use of multiuser MIMO (MU-MIMO) technology, which supports up to four data streams, allowing that many endpoints to simultaneously use a channel. Another benefit of this newer standard is its support for *beamforming*, which is the shaping of radio signals to improve their performance in specific directions. In simple terms, this means that 802.11ac is better able to maintain high data rates at longer ranges than its predecessors.

802.11ax (Wi-Fi 6)

Higher data rates are not always the best way to solve problems, and in the race for faster standards, we took a bunch of shortcuts that made them inefficient. The 802.11ax standard aims to address efficiency rather than faster speeds. A significant improvement it has is a new multiuser OFDM technology that replaces the single-user focused technology used in the 802.11a/g/n/ac standards. This means that multiple stations get to use available channels much more efficiently. In addition, the new standard doubles the number of streams supported by MU-MIMO, which means more stations can use it at the same time. These and many other improvements make 802.11ax much faster and better able to handle very crowded environments.

Other Wireless Network Standards

So far, we've focused pretty heavily on radio-based WLANs. There are other wireless network standards that you should know, at least at a superficial level. These include light-based WLANs and radio-based MANs and PANs. We describe the most important of these standards in the following sections.

Li-Fi

Li-Fi is a wireless networking technology that uses light rather than radio waves to transmit and receive data. It is essentially Wi-Fi using lights instead of radios. You can also think of it as fiber-optic communications without the fiber (i.e., over free space). It turns out that light, like radio, is an electromagnetic wave. The difference is that light is on a much higher frequency range and, thus, can carry significantly more information, at least in theory. Imagine if every light fixture in your home or workplace was able to modulate data onto the light it generates, while your computing device (laptop, smartphone, or whatever) could sense it and use its own light source (maybe the flash on your smartphone) to send data back to the light bulb. Because of the frequencies involved, our eyes are not able to perceive the tiny fluctuations in frequency. Besides, Li-Fi can work over infrared light too, which we can't see anyway.

One of the key benefits of Li-Fi (besides speed and ubiquity) is that it is very constrained to a particular space. Each light bulb has a cone of illumination within which it communicates with specific devices. You don't have to worry about an attacker with a sophisticated antenna picking up your signals a mile away. You can also be pretty confident of who you are communicating with because they have to be right there under the light source. These relatively small areas of service (by a given light source) are called *attocells*. The prefix atto- means quintillionth (which is a pretty small number), but, importantly, it's the next prefix down after femto-, as in *femtocells*, which are tiny cells used in some cellular networks.

At the time of this writing, Li-Fi technology is in its infancy but holds great promise. There are still many challenges to overcome, including co-channel interference (where multiple light sources overlap each other), roaming (seamlessly transferring a communications channel to an adjacent attocell or to an RF-based system if the user wanders out of the supported area), and endpoint interface devices (the sensors and light sources that would have to be built into each laptop, smartphone, etc.). Still, the benefits are many. Apart from the ones mentioned in the previous paragraph, Li-Fi promises to support much higher densities of endpoints, with much lower latencies, and in places where RF can be problematic (e.g., healthcare facilities, aircraft cabins, and power plants).

802.16

IEEE standard 802.16 is a MAN wireless standard that allows for wireless traffic to cover a much wider geographical area, where stations can be as far as 70 km apart. It uses some of the same bands as WLAN standards, specifically 2.4 GHz and 5 GHz, but uses up to 256 subcarriers with variable data rates to efficiently handle lots of traffic across large distances. This technology is also referred to as *broadband* wireless access.

A commercial technology that is based on 802.16 is WiMAX, which was widely touted as a replacement for second-generation (2G) digital cellular networks, particularly in rural areas. While this did not happen across the board (it largely lost out to Long Term Evolution or LTE), 802.16, and WiMAX in particular, remains in widespread use, especially outside the United States. A common implementation of 802.16 technology is shown in Figure 12-3.

NOTE IEEE 802.16 is a standard for vendors to follow to allow for interoperable broadband wireless connections. IEEE does not test for compliance to this standard. The WiMAX Forum runs a certification program that is intended to guarantee compliance with the standard and interoperability with equipment between vendors.

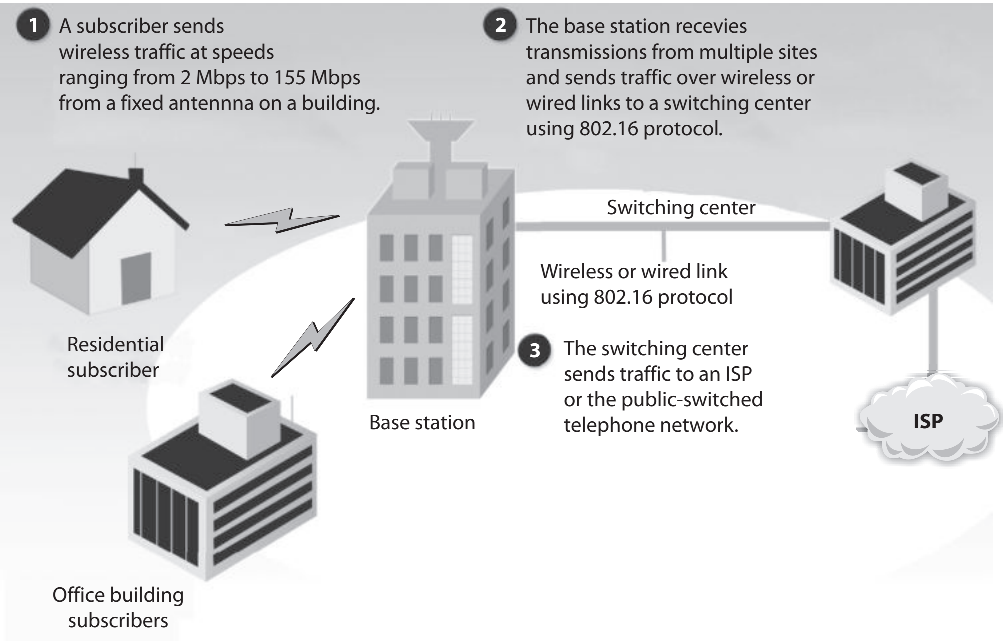

1 A subscriber sends wireless traffic at speeds ranging from 2 Mbps to 155 Mbps from a fixed antennna on a building.
2 The base station recevies transmissions from multiple sites and sends traffic over wireless or wired links to a switching center using 802.16 protocol.
Switching center
Wireless or wired link using 802.16 protocol
3 The switching center sends traffic to an ISP or the public-switched telephone network.
Residential subscriber
Base station
ISP
Office building subscribers

Figure 12-3 Broadband wireless in MAN

802.15.4

The IEEE 802.15.4 standard deals with a much smaller geographical network, which is referred to as a *wireless personal area network (WPAN)*. This technology allows for connectivity to take place among "disadvantaged" devices, which are the ubiquitous low-cost, low-data-rate, low-power, extended-life ones such as the embedded devices introduced in Chapter 7. For example, if you are using active radio frequency identification (RFID) or Industrial Internet of Things (IIoT) devices, odds are that you are using 802.15.4. This standard is optimized for situations in which machines communicate directly with other machines over relatively short distances (typically no more than 100 meters). For this reason, this standard is a key enabler of the Internet of Things (IoT), in which everything from your thermostat to your door lock is (relatively) smart and connected.

The 802.15.4 standard defines the physical (PHY) layer and Media Access Control (MAC) sublayer of the data link layer in the OSI model. At the physical layer, it uses DSSS. For MAC, it uses CSMA-CA. In terms of topology, this standard supports star, tree, and mesh networks. The catch is that, regardless of the topology, 802.15.4 requires a full-function device (FFD) that acts as a central node for the network (even if it is not logically or physically placed at its center). This central device is called the *coordinator* for one or more connected reduced-function devices (RFDs). This makes a lot of sense in a star or tree topology, where you have a regular computer as the hub or root node. It might be a bit less intuitive when you think of mesh networks such as you would find in a smart home network, but we'll get into that when we discuss ZigBee in the next section.

There are multiple extensions to the base 802.15.4 standard that optimize it for specific geographic regions or applications. You may come across the following:

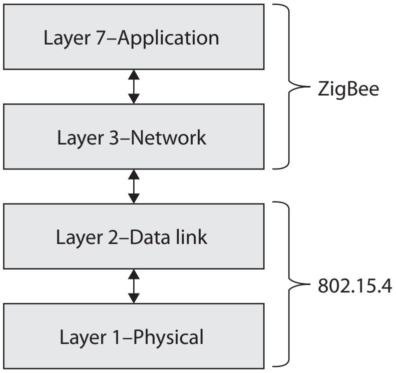
802.15.4c	For use in China
802.15.4d	For use in Japan
802.15.4e	For industrial applications
802.15.4f	For active (i.e., battery powered) radio frequency identification (RFID)
802.15.4g	For smart utility networks (SUNs)

Because this standard was intended to support embedded devices in close proximity to each other, the typical range is only about 10 meters (though it could reach 1 km in optimal conditions) and the data rates are quite low. While nodes frequently communicate at the maximum rate of 250 Kbps, there are also lower rates of 100, 20, and even 10 Kbps for smaller devices that have to last a long time on small batteries. Despite the low data rates, devices that implement this standard are able to support real-time applications (i.e., those that require extremely low latencies) through the use of Guaranteed Time Slot (GTS) reservations. Note that when a GTS is used, the channel access technique used has to be time division multiple access (TDMA) instead of the more typical CSMA/CA. TDMA is a technique that divides each communications channel into multiple time slots to increase the data rates by taking advantage of the fact that not every station will be transmitting all the time.

Security-wise, 802.15.4 implements access control lists (ACLs) by default, so nodes can decide whether to communicate with other nodes based on their claimed physical address. Keep in mind, however, that spoofing a physical address is trivial. The standard also offers (but does not require) two other security mechanisms that you should know. The first is support for symmetric key encryption using the Advanced Encryption Standard (AES) with 128-bit keys, used to protect message confidentiality and integrity. The second is a frame counter feature that protects against replay attacks by tracking the last message received from another node and ensuring a new message is more recent than it.

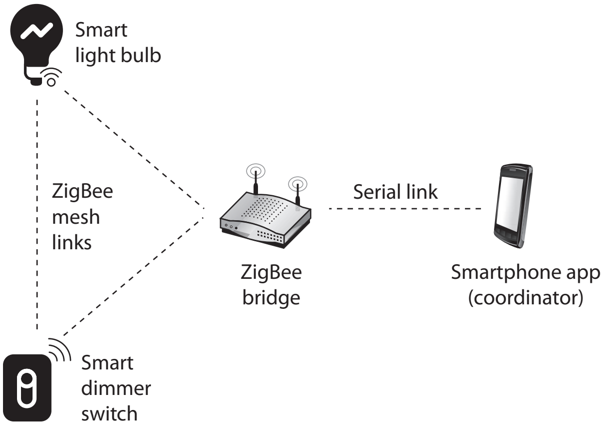
ZigBee

ZigBee is one of the most popular standards based on IEEE 802.15.4. It sits right on top of the layer 1 and layer 2 services provided by 802.15.4 and adds networking and application layer support.



ZigBee is intended to be simpler and cheaper than most WPAN protocols and is very popular in the embedded device market. You can find ZigBee in a variety of home automation, industrial control, medical, and sensor network applications. Figure 12-4

Figure 12-4 ZigBee in a smart home



shows a typical implementation of this standard for controlling lights in a smart home. All light bulbs and switches are able to talk to each other either directly, through the bridge, or by relaying traffic for destination nodes that are too far from the source node. Note that the connection between the bridge and the controller is over a serial link, but this could be implemented as a connection over Wi-Fi, Bluetooth, or any other means.

Because ZigBee is meant to be used in embedded devices that don't have (and can't afford) a bunch of operating system overhead, it assumes what is called an *open trust model*. This means that all applications within a device trust each other, which indirectly extends to all devices in a network as well. It also means that perimeter protection is absolutely critical. This protection should be implemented both physically and logically. At the physical level, ZigBee devices should be tamper-resistant to prevent attackers from simply reading encryption keys or otherwise gaining physical control of a node and using it as a beachhead for future attacks. At the logical level, it means controlling access to the network, which is done primarily through key management.

The ZigBee standard defines three different 128-bit symmetric keys:

- **Network key** Shared by all nodes to support broadcasts
- **Link key** Unique for each pair of connected devices and used for unicasts
- **Master key** Unique for each pair of connected devices and used for the Symmetric-Key Key Establishment (SKKE) protocol from which the other keys are derived

Since embedded devices oftentimes lack a user interface, the ZigBee standard allows multiple ways to distribute and subsequently manage these keys. The most secure way is based on a centralized security model in which the coordinator node acts as a Trust Center. This node is responsible for authenticating new devices that attempt to join the network and then securely sending them the keys they need. To facilitate this, manufacturers of ZigBee devices can install unique certificates at the factory, which are then used by the Trust Center to authenticate them and distribute keys using the Certificate-Based Key Establishment (CBKE) protocol. This is not all that common of an approach apart from high-security commercial systems. More commonly, manufacturers install a unique key in each device, which is then used by the SKKE protocol to derive keys, much like we would do using the Diffie-Hellman algorithm we covered in Chapter 8. This second approach is less secure, doesn't require a Trust Center, and is typical in consumer systems.

EXAM TIP ZigBee is most secure when a coordinator node acts as a Trust Center.

Bluetooth

The *Bluetooth wireless* technology has a 1- to 3-Mbps transfer rate and works in a range of approximately 1, 10, or 100 meters. It was originally invented as an alternative to connecting devices using cables. Unsurprisingly, its most common application today is in cordless headsets for smartphones. However, the technology has plenty of other uses.

If you have a cell phone and a tablet that are both Bluetooth-enabled and both have calendar functionality, you could have them update each other without any need to connect them physically. If you added some information to your cell phone contacts list and task list, for example, you could just place the phone close to your tablet. The tablet would sense that the other device is nearby, and it would then attempt to set up a network connection with it. Once the connection was made, synchronization between the two devices would take place, and the tablet would add the new contacts list and task list data. Bluetooth works in a portion of the frequency band used by 802.11 devices (2.4 GHz).

In early versions of Bluetooth, real security risks existed due to protocol vulnerabilities, but they have been largely mitigated. Still, as with any other technology, it is possible for attackers to compromise the confidentiality, integrity, or availability of Bluetooth devices. One attack type to which these devices are vulnerable is called *Bluesnarfing*, which is the unauthorized access from a wireless device through a Bluetooth connection. This allows attackers to read, modify, or delete calendar events, contacts, e-mails, text messages, and so on. While recent versions of the Bluetooth standard make this much harder, it is still possible to trick unwary users into accepting an attacker's connection attempts.

Another attack type that Bluetooth is vulnerable to is referred to as *Bluejacking*. In this attack, someone sends an unsolicited message to a device that is Bluetooth-enabled. Bluejackers look for a receiving device (phone, tablet, laptop) and then send a message to it. The countermeasure is to put the Bluetooth-enabled device into nondiscoverable mode so others cannot identify this device in the first place. If you receive some type of message this way, just look around you. Bluetooth only works within a 10-meter distance, so it is coming from someone close by.

Other Important Standards

The wireless standards we've covered so far cover the ways in which devices connect to each other and send data over the radio links they create. Over the years, we've discovered that there are a bunch of other features we want in our wireless networks, regardless of the communications standards being used by the radios themselves. These include Quality of Service (QoS), roaming, and spectrum management issues. Let's take a look at another set of standards you should know.

802.11e

This standard provides QoS and support of multimedia traffic in wireless transmissions. Voice, streaming video, and other types of time-sensitive applications have a lower tolerance for delays in data transmission. The problem is that the original 802.11 protocol treated all traffic equally. In other words, an e-mail message that could safely take minutes to get through had exactly the same priority as a video packet whose tolerable latency is measured in fractions of a second. To address this, the 802.11e standard defines four access categories (ACs) in increasing priority: background, best effort, video, and voice. This QoS provides the capability to prioritize traffic and affords guaranteed delivery. This standard and its capabilities have opened the door to allow many different types of data to be transmitted over wireless connections.

802.11f

When a user moves around in a WLAN, her wireless device often needs to communicate with different APs. An AP can cover only a certain distance, and as the user moves out of the range of the first AP, another AP needs to pick up and maintain her signal to ensure she does not lose network connectivity. This is referred to as *roaming*, and for this to happen seamlessly, the APs need to communicate with each other. If the second AP must take over this user's communication, it needs to be assured that this user has been properly authenticated and must know the necessary settings for this user's connection. This means the first AP needs to be able to convey this information to the second AP. The conveying of this information between the different APs during roaming is what 802.11f deals with. The process of transferring between one AP and another is sometimes called *handoff*. It outlines how this information can be properly shared.

802.11h

Because the ISM bands are unlicensed, devices that operate in them are expected to deal well with interference from other devices. This was all good and well before the explosion of WLANs and Bluetooth devices, but quickly became an issue as crowding increased. To make things worse, the 5-GHz band is used not only for Wi-Fi but also for certain radar and satellite communications systems. In this increasingly busy portion of the spectrum, something had to be done to deal with interference.

The 802.11h standard was originally developed to address these issues in Europe, where interference in the 5-GHz band was particularly problematic. However, the techniques it implements are applicable in many countries around the world. Two specific technologies included in the standard are *Dynamic Frequency Selection (DFS)* and *Transmit Power Control (TPC)*. DFS is typically implemented in the WLAN AP and causes it to automatically select channels that have less interference, particularly from radars. TPC causes any device to automatically reduce its power output when it detects interference from other networks.

802.11j

Japan regulates its radio spectrum differently than many other countries, particularly in the 4.9- and 5-GHz bands. Specifically, Japan uses different frequencies, radio channel widths, and wireless operating settings. In order to allow international devices to be interoperable in Japan, the IEEE developed the 802.11j standard. The need for this standard underscores the fact that each country has the sovereign right to regulate its radio spectrum as it sees fit.

Evolution of WLAN Security

To say that security was an afterthought in the first WLANs would be a remarkable understatement. As with many new technologies, wireless networks were often rushed to market with a focus on functionality, even if that sometimes came at the expense of security. Over time, vendors and standards bodies caught on and tried to correct these omissions. While we have made significant headway in securing our wireless networks, as security professionals we must acknowledge that whenever we transmit anything over the electromagnetic spectrum, we are essentially putting our data in the hands (or at least within the grasp) of our adversaries.

802.11

When WLANs were being introduced, there was industry-wide consensus that some measures would have to be taken to assure users that their data (now in the air) would be protected from eavesdropping to the same degree that data on a wired LAN was already protected. This was the genesis of *Wired Equivalent Privacy (WEP)*. This first WLAN standard, codified as part of the original IEEE 802.11, had a tremendous number of security flaws. These were found within the core standard itself, as well as in different implementations of this standard. Before we delve into these deficiencies, it will be useful to spend a bit of time with some of the basics of 802.11.

EXAM TIP If you ever come across WEP in the context of wireless security, you know it's the wrong answer (unless the question is asking for the least secure standard).

The wireless devices using this protocol can authenticate to the AP in two main ways: *open system authentication (OSA)* and *shared key authentication (SKA)*. OSA does not require the wireless device to prove to the AP it has a specific cryptographic key to allow for authentication purposes. In many cases, the wireless device needs to provide only the correct SSID value. In OSA implementations, all transactions are in cleartext because no encryption is involved. So an intruder can sniff the traffic, capture the necessary steps of authentication, and walk through the same steps to be authenticated and associated to an AP.

When an AP is configured to use SKA, the AP sends a random value to the wireless device. The device encrypts this value with a preshared key (PSK) and returns it. The AP decrypts and extracts the response, and if it is the same as the original value, the device is authenticated. In this approach, the wireless device is authenticated to the network by proving it has the necessary encryption key. The PSK, commonly known as the Wi-Fi password, is a 64- or 128-bit key.

The three core deficiencies with WEP are the use of static encryption keys, the ineffective use of initialization vectors, and the lack of packet integrity assurance. The WEP protocol uses the RC4 algorithm, which is a stream-symmetric cipher. *Symmetric* means the sender and receiver must use the exact same key for encryption and decryption purposes. The 802.11 standard does not stipulate how to update these keys through an automated process, so in most environments, the RC4 symmetric keys are never changed out. And usually all of the wireless devices and the AP share the exact same key. This is like having everyone in your company use the exact same password. Not a good idea. So that is the first issue—static WEP encryption keys on all devices.

The next flaw is how initialization vectors (IVs) are used. An IV is a numeric seeding value that is used with the symmetric key and RC4 algorithm to provide more randomness to the encryption process. Randomness is extremely important in encryption because any patterns can give the bad guys insight into how the process works, which may allow them to uncover the encryption key that was used. The key and 24-bit IV value are inserted into the RC4 algorithm to generate a key stream. The values (1's and 0's) of the key stream are XORed with the binary values of the individual packets. The result is ciphertext, or encrypted packets.

In most WEP implementations, the same IV values are used over and over again in this process, and since the same symmetric key (or shared secret) is generally used, there is no way to provide effective randomness in the key stream that is generated by the algorithm. The appearance of patterns allows attackers to reverse-engineer the process to uncover the original encryption key, which can then be used to decrypt future encrypted traffic.

So now we are onto the third mentioned weakness, which is the integrity assurance issue. WLAN products that use only the 802.11 standard introduce a vulnerability that is not always clearly understood. An attacker can actually change data within the wireless packets by flipping specific bits and altering the Integrity Check Value (ICV) so the receiving end is oblivious to these changes. The ICV works like a cyclic redundancy check (CRC) function; the sender calculates an ICV and inserts it into a frame's header. The receiver calculates his own ICV and compares it with the ICV sent with the frame. If the ICVs are the same, the receiver can be assured that the frame was not modified during transmission. If the ICVs are different, it indicates a modification did indeed take place and thus the receiver discards the frame. In WEP, there are certain circumstances in which the receiver cannot detect whether an alteration to the frame has taken place; thus, there is no true integrity assurance.

So the problems identified with the 802.11 standard include poor authentication, static WEP keys that can be easily obtained by attackers, IV values that are repetitive and do not provide the necessary degree of randomness, and a lack of data integrity. The next section describes the measures taken to remedy these problems.

NOTE 802.11 and WEP were deprecated years ago, are inherently insecure, and should not be used.

802.11i

IEEE came out with a standard in 2004 that deals with the security issues of the original 802.11 standard, which is called IEEE 802.11i or *Wi-Fi Protected Access 2 (WPA2)*. Why the number 2? Because while the formal standard was being ratified by the IEEE, the Wi-Fi Alliance pushed out WPA (the first one) based on the draft of the standard. For this reason, WPA is sometimes referred to as the *draft* IEEE 802.11i. This rush to push out WPA required the reuse of elements of WEP, which ultimately made WPA vulnerable to some of the same attacks that doomed its predecessor. Let's start off by looking at WPA in depth, since this protocol is still in use despite its weaknesses.

WPA employs different approaches that provide much more security and protection than the methods used in the original 802.11 standard. For starters, the PSK size was increased to 256 bits and is salted with the SSID of the WLAN to make it harder to crack. This is good, but the greatest enhancement of security is accomplished through specific protocols, technologies, and algorithms. The first protocol is *Temporal Key Integrity Protocol (TKIP)*, which is backward-compatible with the WLAN devices based upon the original 802.11 standard. TKIP actually works with WEP by feeding it keying material, which is data to be used for generating new dynamic keys. TKIP generates a new key for every frame that is transmitted. These changes constitute the variety of this standard known as WPA Personal, which is geared at consumers.

NOTE TKIP was developed by the IEEE 802.11i task group and the Wi-Fi Alliance. The goal of this protocol was to increase the strength of WEP or replace it fully without the need for hardware replacement. TKIP provides a key mixing function, which allows the RC4 algorithm to provide a higher degree of protection. It also provides a sequence counter to protect against replay attacks and implements a message integrity check mechanism.

There is also a more robust version called WPA Enterprise. The main difference is that it also integrates 802.1X port authentication and Extensible Authentication Protocol (EAP) authentication methods. The use of the 802.1X technology (which we'll discuss in its own section shortly) provides access control by restricting network access until full authentication and authorization have been completed, and provides a robust authentication framework that allows for different EAP modules to be plugged in. These two technologies (802.1X and EAP) work together to enforce mutual authentication between the wireless device and authentication server. So what about the static keys, IV value, and integrity issues?

TKIP addresses the deficiencies of WEP pertaining to static WEP keys and inadequate use of IV values. Two hacking tools, AirSnort and WEPCrack, can be used to easily crack WEP's encryption by taking advantage of these weaknesses and the ineffective use of the key scheduling algorithm within the WEP protocol. If a company is using products that implement only WEP encryption and is not using a third-party encryption solution (such as a VPN), these programs can break its encrypted traffic within minutes. There is no "maybe" pertaining to breaking WEP's encryption. Using these tools means it will be broken whether a 40-bit or 128-bit key is being used—it doesn't matter. This is one of the most serious and dangerous vulnerabilities pertaining to the original 802.11 standard.

The use of TKIP provides the ability to rotate encryption keys to help fight against these types of attacks. The protocol increases the length of the IV value and ensures that every frame has a different IV value. This IV value is combined with the transmitter's MAC address and the original WEP key, so even if the WEP key is static, the resulting encryption key will be different for every frame. (WEP key + IV value + MAC address = new encryption key.) So what does that do for us? This brings more randomness to the encryption process, and it is randomness that is necessary to properly thwart cryptanalysis and attacks on cryptosystems. The changing IV values and resulting keys make the resulting key stream less predictable, which makes it much harder for the attacker to reverse-engineer the process and uncover the original key.

TKIP also deals with the integrity issues by using a message integrity check (MIC) instead of an ICV function. If you are familiar with a message authentication code

(MAC) function, this is the same thing. A symmetric key is used with a hashing function, which is similar to a CRC function but stronger. The use of a MIC instead of an ICV function ensures the receiver will be properly alerted if changes to the frame take place during transmission. The sender and receiver calculate their own separate MIC values. If the receiver generates a MIC value different from the one sent with the frame, the frame is seen as compromised and it is discarded.

The types of attacks that have been carried out on WEP devices and networks that just depend upon WEP are numerous and unnerving. Wireless traffic can be easily sniffed, data can be modified during transmission without the receiver being notified, rogue APs can be erected (which users can authenticate to and communicate with, not knowing it is a malicious entity), and encrypted wireless traffic can be decrypted quickly and easily. Unfortunately, these vulnerabilities usually provide doorways to the actual wired network where the more destructive attacks can begin.

The full 802.11i (WPA2) has a major advantage over WPA by providing encryption protection with the use of the AES algorithm in counter mode with CBC-MAC (CCM), which is referred to as the Counter Mode Cipher Block Chaining Message Authentication Code Protocol (CCM Protocol or CCMP). AES is a more appropriate algorithm for wireless than RC4 and provides a higher level of protection. WPA2 defaults to CCMP, but can switch down to TKIP and RC4 to provide backward compatibility with WPA devices and networks.

802.11w

WPA2 was a huge step forward for WLAN security because it provided effective encryption for most wireless traffic. However, there are certain frames that cannot be encrypted because every station (even those that have not yet joined the network) must be able to receive. These are called *management frames*, and they take care of things like beaconing, association, and authentication. While we can't encrypt them, we can take measures to ensure their integrity. The IEEE 802.11w standard provides Management Frame Protection (MFP) that prevents certain types of attacks, such as replay and denial-of-service (DoS) attacks.

A particularly problematic type of DoS attack on WLANs is called a deauthentication (or deauth) attack and it exploits a feature of Wi-Fi that allows WAPs to disconnect rogue devices by sending a deauthentication management frame. You can see how, in an environment without MFP, it would be trivial for an attacker to spoof such messages, claiming to be the real WAP. 802.11w solves this problem for WLANs that are not yet on WPA3.

WPA3

Like any other security mechanism, WPA2 began to crack under intensifying attacks. By 2018, the Wi-Fi Alliance decided that a new approach was needed. The result is WPA3, which is not directly equivalent to any IEEE standard, though it does require 802.11w to protect management frames. Like its predecessor WPA2, WPA3 comes in two flavors: Personal and Enterprise.

WPA3 Personal is aimed at the consumer market and tries to make security transparent to the average user. One of the most important innovations of this standard is that it allows users to choose passwords that, though they might be easily guessable, still provide adequate security. This is done through Simultaneous Authentication of Equals (SAE), which is defined in IEEE 802.11s, instead of relying on WPA2's preshared keys. SAE uses the Diffie-Hellman key exchange method but adds an authentication element based on the (potentially weak) password. The result is a secret session key that is remarkably resistant to password-cracking attempts.

WPA3 Enterprise is similar to its predecessor (WPA2 Enterprise) but makes use of stronger cryptography. It does this by restricting the allowed algorithms to a handful of strong ones that use 192-bit keys. It also requires certificates on both the AP and the wireless device for mutual authentication. The challenge with deploying WPA3 is that many older wireless interfaces, particularly those on most embedded devices, cannot support it, which means you may have to upgrade many (or all) of your endpoints.

802.1X

The 802.11i standard can be understood as three main components in two specific layers. The lower layer contains the improved encryption algorithms and techniques (TKIP and CCMP), while the layer that resides on top of it contains 802.1X. They work together to provide more layers of protection than the original 802.11 standard.

The 802.1X standard is a port-based network access control protocol that ensures a user cannot make a full network connection until he is properly authenticated. This means a user cannot access network resources and no traffic is allowed to pass, other than authentication traffic, from the wireless device to the network until the user is properly authenticated. An analogy is having a chain on your front door that enables you to open the door slightly to identify a person who knocks before you allow him to enter your house.

NOTE 802.1X is not a wireless protocol. It is an access control protocol that can be implemented on both wired and wireless networks.

By incorporating 802.1X, the new standard allows for the user to be authenticated, whereas using only WPA provides *system* authentication. User authentication provides a higher degree of confidence and protection than system authentication. The 802.1X technology actually provides an authentication framework and a method of dynamically distributing encryption keys. The three main entities in this framework are the supplicant (wireless device), the authenticator (AP), and the authentication server (usually a RADIUS server).

The AP controls all communication and allows the wireless device to communicate with the authentication server and wired network only when all authentication steps are completed successfully. This means the wireless device cannot send or receive HTTP, DHCP, SMTP, or any other type of traffic until the user is properly authorized. WEP does not provide this type of strict access control.

Another disadvantage of the original 802.11 standard is that mutual authentication is not possible. When using WEP alone, the wireless device can authenticate to the AP, but the authentication server is not required to authenticate to the wireless device. This means a rogue AP can be set up to capture users' credentials and traffic without the users being aware of this type of attack. 802.11i deals with this issue by using EAP. EAP allows for mutual authentication to take place between the authentication server and wireless device and provides flexibility in that users can be authenticated by using passwords, tokens, one-time passwords, certificates, smart cards, or Kerberos. This allows wireless users to be authenticated using the current infrastructure's existing authentication technology. The wireless device and authentication server that are 802.11i-compliant have different authentication modules that plug into 802.1X to allow for these different options. So, 802.1X provides the framework that allows for the different EAP modules to be added by a network administrator. The two entities (supplicant and authenticator) agree upon one of these authentication methods (EAP modules) during their initial handshaking process.

The 802.11i standard does not deal with the full protocol stack, but addresses only what is taking place at the data link layer of the OSI model. Authentication protocols reside at a higher layer than this, so 802.11i does not specify particular authentication protocols. The use of EAP, however, allows different protocols to be used by different vendors. For example, Cisco uses a purely password-based authentication framework called Lightweight Extensible Authentication Protocol (LEAP). Other vendors, including Microsoft, use EAP and Transport Layer Security (EAP-TLS), which carries out authentication through digital certificates. And yet another choice is Protected EAP (PEAP), where only the server uses a digital certificate.

EAP-Tunneled Transport Layer Security (EAP-TTLS) is an EAP protocol that extends TLS. EAP-TTLS is designed to provide authentication that is as strong as EAP-TLS, but it does not require that each user be issued a certificate. Instead, only the authentication servers are issued certificates. User authentication is performed by password, but the password credentials are transported in a securely encrypted tunnel established based upon the server certificates.

If EAP-TLS is being used, the authentication server and wireless device exchange digital certificates for authentication purposes. If PEAP is being used instead, the user of the wireless device sends the server a password and the server authenticates to the wireless device with its digital certificate. In both cases, some type of public key infrastructure (PKI) needs to be in place. If a company does not have a PKI currently implemented, it can be an overwhelming and costly task to deploy a PKI just to secure wireless transmissions.

When EAP-TLS is being used, the steps the server takes to authenticate to the wireless device are basically the same as when a TLS connection is being set up between a web server and web browser. Once the wireless device receives and validates the server's digital certificate, it creates a master key, encrypts it with the server's public key, and sends it over to the authentication server. Now the wireless device and authentication server have a master key, which they use to generate individual symmetric session keys. Both entities use these session keys for encryption and decryption purposes, and it is the use of these keys that sets up a secure channel between the two devices.

Organizations may choose to use PEAP instead of EAP-TLS because they don't want the hassle of installing and maintaining digital certificates on every wireless device.

Before you purchase a WLAN product, you should understand the requirements and complications of each method to ensure you know what you are getting yourself into and if it is the right fit for your environment.

A large concern with any WLANs using just WEP is that if individual wireless devices are stolen, they can easily be authenticated to the wired network. 802.11i has added steps to require the user to authenticate to the network instead of just requiring the wireless device to authenticate. By using EAP, the user must send some type of credential set that is tied to his identity. When using only WEP, the wireless device authenticates itself by proving it has a symmetric key that was manually programmed into it. Since the user does not need to authenticate using WEP, a stolen wireless device can allow an attacker easy access to your precious network resources.

The Answer to All Our Prayers?

So, does the use of EAP, 802.1X, AES, and TKIP result in secure and highly trusted WLAN implementations? Maybe, but we need to understand what we are dealing with here. TKIP was created as a quick fix to WEP's overwhelming problems. It does not provide an overhaul for the wireless standard itself because WEP and TKIP are still based on the RC4 algorithm, which is not the best fit for this type of technology. The use of AES is closer to an actual overhaul, but it is not backward-compatible with the original 802.11 implementations. In addition, we should understand that using all of these new components and mixing them with the current 802.11 components will add more complexity and steps to the process. Security and complexity do not usually get along. The highest security is usually accomplished with simplistic and elegant solutions to ensure all of the entry points are clearly understood and protected. These newer technologies add more flexibility to how vendors can choose to authenticate users and authentication servers, but can also bring us interoperability issues because the vendors will not all choose the same methods. This means that if an organization buys an AP from company A, then the wireless cards the organization buys from companies B and C may not work seamlessly.

So, does that mean all of this work has been done for naught? No. 802.11i provides much more protection and security than WEP ever did. The working group has had very knowledgeable people involved and some very large and powerful companies aiding in the development of these new solutions. But the customers who purchase these new products need to understand what will be required of them *after* their purchase. For example, with the use of EAP-TLS, each wireless device needs its own digital certificate. Are your current wireless devices programmed to handle certificates? How will the certificates be properly deployed to all the wireless devices? How will the certificates be maintained? Will the devices and authentication server verify that certificates have not been revoked by periodically checking a certificate revocation list (CRL)? What if a rogue authentication server or AP was erected with a valid digital certificate? The wireless device would just verify this certificate and trust that this server is the entity it is supposed to be communicating with.

Today, WLAN products are being developed following the stipulations of this 802.11i wireless standard. Many products will straddle the fence by providing TKIP for backward-compatibility with current WLAN implementations and AES for organizations that are just now thinking about extending their current wired environments with a wireless component. Before buying wireless products, customers should review the Wi-Fi Alliance's certification findings, which assess systems against the 802.11i proposed standard.

Best Practices for Securing WLANs

There is no silver bullet to protect any of our devices or networks. That being said, there are a number of things we can do that will increase the cost of the attack for the adversary. Some of the best practices pertaining to WLAN implementations are as follows:

- Change the default SSID. Each AP comes with a preconfigured default SSID value that may reveal the manufacturer and even model number, which may advertise systems with known vulnerabilities.
- Implement WPA3 Enterprise to provide centralized user authentication (e.g., RADIUS, Kerberos). Before users can access the network, require them to authenticate.
- Use separate VLANs for each class of users, just as you would on a wired LAN.
- If you must support unauthenticated users (e.g., visitors), ensure they are connected to an untrusted VLAN that remains outside your network's perimeter.
- Deploy a wireless intrusion detection system (WIDS).
- Physically put the AP at the center of the building to limit how far outside the facility the signal will reach (and be reachable). The AP has a specific zone of coverage it can provide.
- Logically put the AP in a DMZ with a firewall between the DMZ and internal network. Allow the firewall to investigate the traffic before it gets to the wired network.
- Implement VPN for wireless devices to use. This adds another layer of protection for data being transmitted.
- Configure the AP to allow only known MAC addresses into the network. Allow only known devices to authenticate. But remember that these MAC addresses are sent in cleartext, so an attacker could capture them and masquerade himself as an authenticated device.
- Carry out penetration tests on the WLAN. Use the tools described in this section to identify APs and attempt to break the current encryption scheme being used.

Mobile Wireless Communication

Mobile wireless has now exploded into a trillion-dollar industry, with over 14 billion devices worldwide, fueled by a succession of new technologies and by industry and international standard agreements. So what is a mobile phone anyway? It is a device that can send voice and data over wireless radio links. It connects to a cellular network, which is connected to the public switched telephone network (PSTN). So instead of needing a physical cord and connection that connects your phone and the PSTN, you have a device that allows you to indirectly connect to the PSTN as you move around a wide geographic area.

A cellular network distributes radio signals over delineated areas, called *cells*. Each cell has at least one fixed-location transceiver (base station) and is joined to other cells to provide connections over large geographic areas. So as you are talking on your mobile phone and you move out of one cell, the base station in the original cell sends your connection information to the base station in the next cell so that your call is not dropped and you can continue your conversation.

We do not have an infinite number of frequencies to work with when it comes to mobile communication. Millions of people around the world are using their cell phones as you read this. How can all of these calls take place if we only have one set of frequencies to use for such activity? Individual cells can use the same frequency range, as long as they are not right next to each other. So the same frequency range can be used in every other cell, which drastically decreases the amount of ranges required to support simultaneous connections. A rudimentary depiction of a cellular network, in which nonadjacent cells reuse the frequency sets F0, F1, F2, F3, and F4, is shown in Figure 12-5.

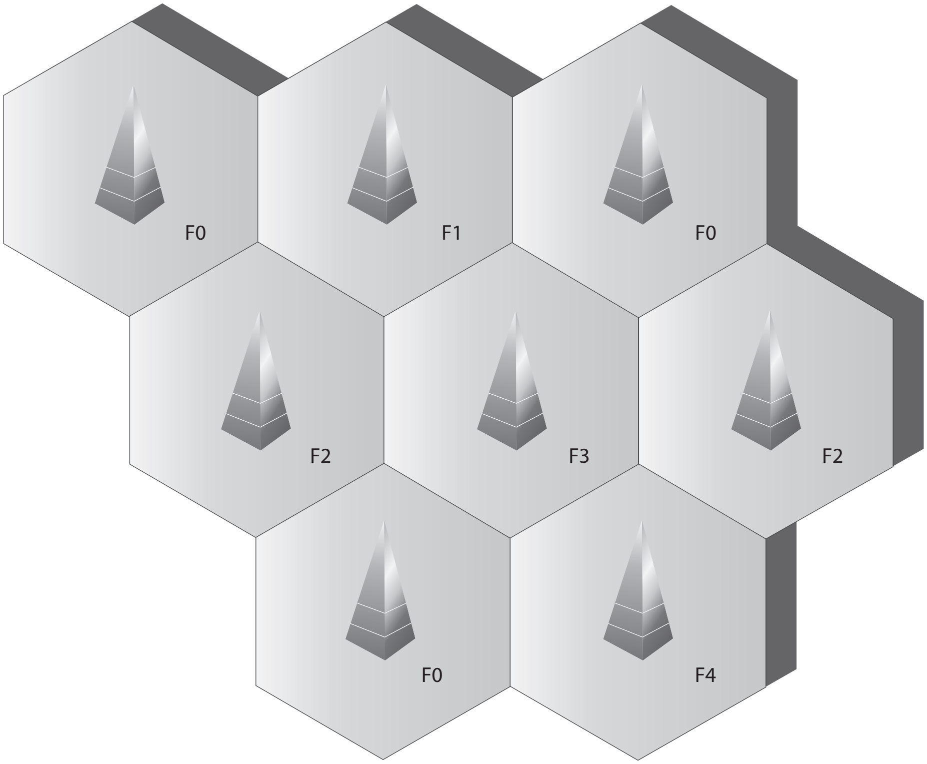


Figure 12-5 Nonadjacent cells can use the same frequency ranges.

Multiple Access Technologies

The industry has had to come up with other ways to allow millions of users to be able to use this finite resource (frequency range) in a flexible manner. Over time, mobile wireless has been made up of progressively more complex and more powerful "multiple access" technologies, listed here:

- Frequency division multiple access (FDMA)
- Time division multiple access (TDMA)
- Code division multiple access (CDMA)
- Orthogonal frequency division multiple access (OFDMA)

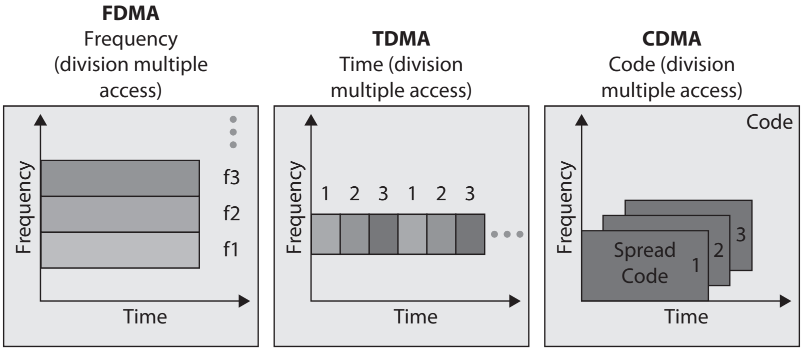
We'll quickly go over the characteristics of each of these technologies because they are the foundational constructs of the various cellular network generations.

Frequency division multiple access (FDMA) was the earliest multiple access technology put into practice. The available frequency range is divided into sub-bands (channels), and one channel is assigned to each subscriber (cell phone). The subscriber has exclusive use of that channel while the call is made, or until the call is terminated or handed off; no other calls or conversations can be made on that channel during that call. Using FDMA in this way, multiple users can share the frequency range without the risk of interference between the simultaneous calls. FDMA was used in the first generation (1G) of cellular networks. Various 1G mobile implementations, such as Advanced Mobile Phone System (AMPS), Total Access Communication System (TACS), and Nordic Mobile Telephone (NMT), used FDMA.

Time division multiple access (TDMA) increases the speed and efficiency of the cellular network by taking the radio frequency spectrum channels and dividing them into time slots. At various time periods, multiple users can share the same channel; the systems within the cell swap from one user to another user, in effect, reusing the available frequencies. TDMA increased speeds and service quality. A common example of TDMA in action is a conversation. One person talks for a time and then quits, and then a different person talks. In TDMA systems, time is divided into frames. Each frame is divided into slots. TDMA requires that each slot's start and end time are known to both the source and the destination. Mobile communication systems such as Global System for Mobile Communication (GSM), Digital AMPS (D-AMPS), and Personal Digital Cellular (PDC) use TDMA.

Code division multiple access (CDMA) was developed after FDMA, and as the term "code" implies, CDMA assigns a unique code to each voice call or data transmission to uniquely identify it from all other transmissions sent over the cellular network. In a CDMA "spread spectrum" network, calls are spread throughout the entire radio frequency band. CDMA permits every user of the network to simultaneously use every channel in the network. At the same time, a particular cell can simultaneously interact with multiple other cells. These features make CDMA a very powerful technology.

It is the main technology for the mobile cellular networks that presently dominate the wireless space.


FDMA
Frequency (division multiple access)
TDMA
Time (division multiple access)
CDMA
Code (division multiple access)
Frequency
f3
f2
f1
Time
Frequency
1 2 3 1 2 3
Time
Code
Frequency
Spread Code
1
2
3
Time

Orthogonal frequency division multiple access (OFDMA) is derived from a combination of FDMA and TDMA. In earlier implementations of FDMA, the different frequencies for each channel were widely spaced to allow analog hardware to separate the different channels. In OFDMA, each of the channels is subdivided into a set of closely spaced orthogonal frequencies with narrow subchannels. Each of the different subchannels can be transmitted and received simultaneously in a multiple input, multiple output (MIMO) manner. The use of orthogonal frequencies and MIMO allows signal processing techniques to reduce the impacts of any interference between different subchannels and to correct for channel impairments, such as noise and selective frequency fading. 4G and 5G require that OFDMA be used.

Generations of Mobile Wireless

Multiple access technology development was driven by the dramatic growth in mobile subscribers worldwide. Mobile wireless technologies have gone through a whirlwind of confusing generations. The first generation (1G) dealt with analog transmissions of voice-only data over circuit-switched networks. This generation provided a throughput of around 19.2 Kbps. The second generation (2G) allows for digitally encoded voice and data to be transmitted between wireless devices, such as cell phones, and content providers. TDMA, CDMA, GSM, and PCS all fall under the umbrella of 2G mobile telephony. This technology can transmit data over circuit-switched networks and supports data encryption, fax transmissions, and short message services (SMSs).

The third-generation (3G) networks became available around the turn of the century. Incorporating FDMA, TDMA, and CDMA, 3G has the flexibility to support a great variety of applications and services. Further, 3G replaced circuit switching with packet switching. Modular in design to allow ready expandability, backward compatibility with 2G networks, and stressing interoperability among mobile systems, 3G services greatly expanded the applications available to users, such as global roaming (without changing one's cell phone or cell phone number), as well as Internet services and multimedia.

In addition, reflecting the ever-growing demand from users for greater speed, latency in 3G networks was much reduced as transmission speeds were enhanced. More enhancements

Mobile Technology Generations

Like many technologies, the mobile communication technology has gone through several different generations.

First generation (1G):

- Analog services
- Voice service only

Second generation (2G):

- Primarily voice, some low-speed data (circuit switched)
- Phones were smaller in size
- Added functionality of e-mail, paging, and caller ID

Generation 2½ (2.5G):

- Higher data rates than 2G
- "Always on" technology for e-mail and pages

Third generation (3G):

- Integration of voice and data
- Packet-switched technology, instead of circuit-switched

Generation 3.5 G (3GPP)

- Higher data rates
- Use of OFDMA technology

Fourth generation (4G)

- Based on an all-IP packet-switched network
- Data exchange at 100 Mbps to 1 Gbps

Fifth generation (5G)

- Higher frequency ranges, which cut down range and make interference a bigger deal
- Data rates of 20 Gbps possible
- Supports dense deployment of high-speed, low-latency services

to 3G networks, often referred to as 3.5G or as mobile broadband, took place under the rubric of the Third Generation Partnership Project (3GPP). 3GPP resulted in a number of new or enhanced technologies. These include Enhanced Data Rates for GSM Evolution (EDGE), High-Speed Downlink Packet Access (HSDPA), CDMA2000, and Worldwide Interoperability for Microwave Access (WiMAX).

At the time of writing, 4th generation (4G) mobile networks are dominant (though, as we're about to see, that's going to change soon). Initially, there were two competing technologies that fell under the umbrella of 4G: Mobile WiMAX and Long-Term Evolution (LTE). Eventually, however, LTE won out and WiMAX is no longer used in mobile wireless networks. (Though, as we've already discussed, WiMAX is still used as an alternative to traditional ISP services in WANs.) A 4G system does not support traditional circuit-switched telephony service as 3G does, but works over a purely packet-based network. 4G devices are IP-based and are based upon OFDMA instead of the previously used multiple carrier access technologies. In theory, 4G devices should be able to reach 2-Gbps data rates, though that is seldom the case in practice.

Fifth generation (5G) is the technology that is all the rage right now. Its biggest advantage, at least from users' perspectives, over 4G is speed. 5G is capable of reaching a whopping 20 Gbps, which puts it in the neighborhood of the latest Wi-Fi 6 standard. What are the drawbacks of 5G? In order to achieve those jaw-dropping speeds, 5G uses higher frequencies that, as we already discussed, have shorter ranges and are more susceptible to interference. This means that carriers will have to put up more cellular towers.

Each of the different mobile communication generations has taken advantage of the improvement of hardware technology and processing power. The increase in hardware has allowed for more complicated data transmission between users and hence the desire for more users to use mobile communications.

Table 12-2 illustrates some of the main features of the 2G through 5G networks. It is important to note that this table does not and cannot easily cover all the aspects of each generation. Earlier generations of mobile communication have considerable variability between countries. The variability was due to country-sponsored efforts before agreed-upon international standards were established. Various efforts between the ITU and countries have attempted to minimize the differences.

NOTE While it would be great if the mobile wireless technology generations broke down into clear-cut definitions, they do not. This is because various parts of the world use different foundational technologies, and there are several competing vendors in the space with their own proprietary approaches.

	2G	3G	4G	5G
Spectrum	1,800 MHz	2 GHz	Various	Various 3–86 GHz
Bandwidth	25 MHz	25 MHz	100 MHz	30–300 MHz
Multiplexing Type	TDMA	CDMA	OFDMA	OFDMA
New Features Introduced	Digital voice, SMS, MMS	Mobile Internet access, video	Mobile broadband, HD video	Ultra-HD and 3D video
Data Rate	115–128 Kbps	384 kbps	100 Mbps (moving) 1 Gbps (stationary)	Up to 10 Gbps
Introduction	1993	2001	2009	2018

Table 12-2 The Different Characteristics of Mobile Technology

Hacking Mobile Phones

2G networks (which are still around, believe it or not) lack the ability to authenticate towers to phones. In other words, an attacker can easily set up a rogue tower with more power than the nearby legitimate ones and cause the target's mobile phone to connect to it. This type of attack allows attackers to intercept all mobile phone traffic. Though 3G and 4G networks corrected this serious vulnerability, it is sometimes still possible to force most phones to switch down to 2G mode by jamming 3G, 4G, and 5G towers. In an effort to maintain some form of connectivity, handsets may then switch down to the vulnerable 2G mode, making the attack possible again.

Devices designed to perform this type of attack are called International Mobile Subscriber Identity (IMSI) catchers. Initially intended for law enforcement and intelligence agency use, IMSI catchers are increasingly available to criminals in the black markets. Moreover, it is possible for anyone to build one of these attack platforms for less than $1,500, as Chris Paget demonstrated at DefCon in 2010. This is yet another example of how backward compatibility can perpetuate vulnerabilities in older protocols.

Satellites

Today, satellites are used to provide wireless connectivity between distant stations. For two different locations to communicate via satellite links, they must be within the satellite's line of sight and *footprint* (area covered by the satellite), which tends to be large even for low Earth orbit satellites. The sender of information (ground station) modulates the data onto a radio signal that is transmitted to the satellite. A transponder on the satellite receives this signal, amplifies it, and relays it to the receiver. The receiver must have a type of antenna—one of those circular, dish-like things we see on top of buildings. The antenna contains one or more microwave receivers, depending upon how many satellites it is accepting data from.

Satellites provide broadband transmission that is commonly used for television channels and Internet access. If a user is receiving TV data, then the transmission is set up as a one-way (broadcast) network. If a user is using this connection for Internet connectivity, then the transmission is set up as a two-way network. The available bandwidth depends upon the antenna and terminal type and the service provided by the service provider. Time-sensitive applications, such as voice and video conferencing, can suffer from the delays experienced as the data goes to and from the satellite.

There are two types of orbits that are commonly used in satellite communications networks: geosynchronous and low Earth. Traditional networks, like the ones that broadcast TV and carry transoceanic data links for the major carriers, orbit at an altitude of 22,236 miles, which means they rotate at the same rate as the Earth does. This is called a *geosynchronous orbit*, and it makes the satellites appear to be stationary over the same spot on the ground. The key benefit is that the ground station antenna doesn't have to move. The main drawbacks are that, with that kind of range, you need a pretty big antenna and have to wait about a second for a radio wave to go up to the satellite and come back to Earth. This latency can create challenges for real-time communications like video conferencing.

Other satellites use a low Earth orbit (LEO), which is typically between 99 and 1,243 miles above the surface of the Earth. This means there is not as much distance between the ground stations and the satellites as in other types of satellites. In turn, this means smaller receivers can be used, which makes LEO satellites ideal for international cellular communication and Internet use. The catch is that the data rates tend to be much smaller than geosynchronous satellites and the service plans are pretty expensive.

In most cases, organizations use a system known as a very small aperture terminal (VSAT), which links a station (such as a remote office) to the Internet through a satellite gateway facility run by a service provider, as shown in Figure 12-6. Alternatively, VSATs can be deployed in stand-alone networks in which the organization also places a VSAT at a central location and has all the remote ones reach into it with no need for a gateway facility. The data rates available can range from a few Kbps to several Mbps. Dropping prices have rendered this technology affordable to many midsized organizations, though it is still far from being inexpensive.

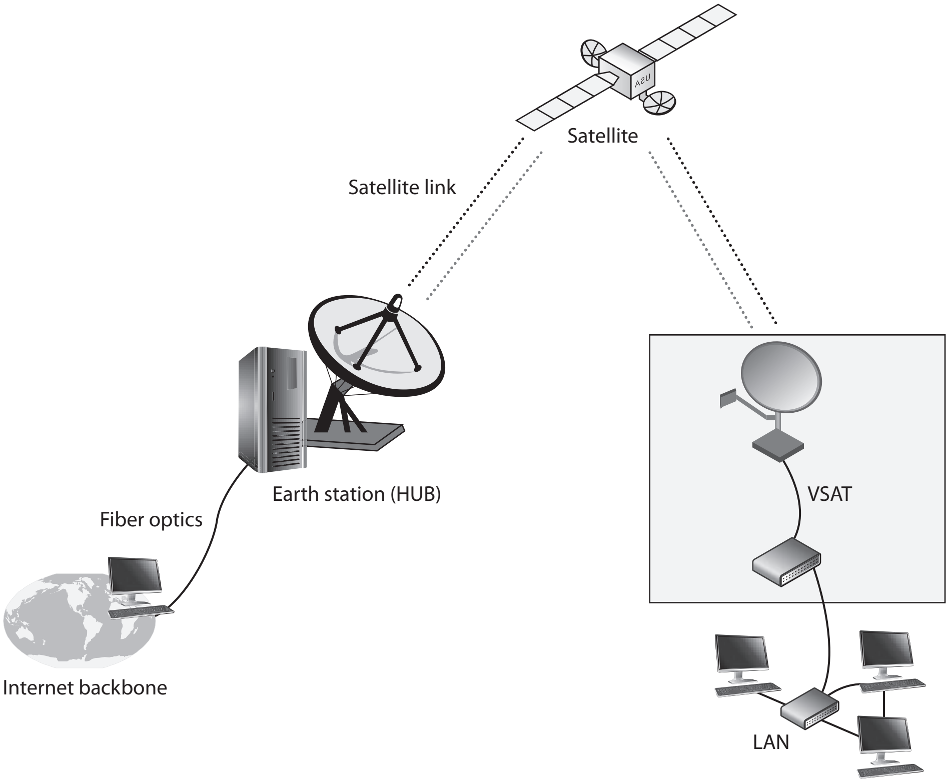


Figure 12-6 Satellite broadband

Chapter Review

Wireless networking is ubiquitous and, over the years, the security community has made great strides to ensure the confidentiality, integrity, and availability of our systems using these technologies. Still, risk can never be driven to zero, and this is particularly true when you transmit into free space, whether you do so using radio or light waves. Best practices for securing wireless networks include using strong cryptography, controlling access, and periodically testing the effectiveness of our controls.

As security professionals, we must always be aware of the myriad of new wireless technologies being developed and sold. For each, we have to compare the benefits (which are always touted by the vendors) to the risks (which may be less obvious and more difficult to identify). The market will constantly push products that promise new features and functionality, even if they come at the cost of security. To be clear, most new technologies incorporate at least some basic security features (and in many cases, advanced security features too), but these are not always implemented in a systematic manner by their adopters. That's where security professionals need to weigh in.

Quick Review

- Wireless communication systems modulate data onto electromagnetic signals like radio and light waves.
- Normally, a higher frequency can carry more data, but over a shorter distance and with more susceptibility to interference.
- Wireless communication systems typically use carrier sense multiple access with collision avoidance (CSMA/CA) as a medium access control (MAC) protocol.
- A radio frequency band is a subset of the radio spectrum designated for a specific use.
- Wi-Fi systems operate in the 2.4-GHz and 5-GHz bands.
- Most wireless communication systems use one of two modulation techniques: spread spectrum or orthogonal frequency division multiplexing (OFDM).
- Spread spectrum modulation techniques include frequency hopping spread spectrum (FHSS) and direct sequence spread spectrum (DSSS).
- DSSS spreads the data being transmitted over a wider spectrum than would otherwise be needed in order to overcome interference and relies on a chip sequence to let receiving stations know how to reconstruct the transmitted data.
- FHSS uses a single subchannel at a time, but rapidly changes channels in a particular hop sequence.
- Wireless local area networks (WLANs) come in two forms: infrastructure and ad hoc.
- Environments can be segmented into different WLANs by using different SSIDs.
- 802.11a provides up to 54 Mbps and operates in the 5-GHz band.

- 802.11b provides a transfer rate of up to 11 Mbps and works in the 2.4-GHz frequency range.
- 802.11g operates in the 2.4-GHz band and supports data rates of up to 54 Mbps.
- 802.11n, also known as Wi-Fi 4, supports throughputs of up to 100 Mbps and works in the 5-GHz band.
- IEEE 802.11ac (Wi-Fi 5) is an extension of 802.11n that increases throughput to 1.3 Gbps and is backward compatible with 802.11a, 802.11b, 802.11g, and 802.11n.
- The 802.11ax standard aims to address efficiency rather than faster speeds.
- Li-Fi is a wireless networking technology that uses light rather than radio waves to transmit and receive data.
- 802.16 is a metropolitan area network (MAN) wireless standard that allows wireless traffic to cover large geographical areas where stations can be as far as 70 km apart, using the 2.4-GHz and 5-GHz bands.
- The 802.15.4 standard defines the physical layer and Media Access Control sublayer of wireless personal area networks (WPANs).
- ZigBee is a standard for layers 3 (network) and 7 (application) that is built on top of 802.15.4 and is most commonly used in Internet of Things (IoT) and Industrial IoT systems.
- Bluetooth is another standard for WPANs, which is most commonly used to replace the cables connecting peripherals to computers and mobile devices.
- The 802.11e standard provides Quality of Service (QoS) and support of multimedia traffic in wireless transmissions.
- 802.11f standardizes the processes by which access points transfer active connections among themselves, enabling users to roam across APs.
- The 802.11h standard was developed to address interference issues in the 5-GHz band, particularly with regard to radar and satellite systems, through Dynamic Frequency Selection (DFS) and Transmit Power Control (TPC) techniques.
- 802.11j is an example of standards that allow common technologies such as WLANs to be employed in countries (in this case Japan) where local regulations conflict with portions of a broader standard (in this case 802.11).
- 802.11 was the original WLAN standard, which included Wired Equivalent Privacy (WEP); it is now obsolete.
- 802.11i defines Wi-Fi Protected Access 2 (WPA2) and is the most common standard in use in WLANs today.
- The IEEE 802.11w standard provides Management Frame Protection (MFP) that prevents certain types of attacks, such as replay and DoS attacks.
- WPA3 was developed by the Wi-Fi alliance (not IEEE) and is quickly replacing WPA2 for both personal and enterprise use.

- 802.1X is an access control protocol that can be implemented on both wired and wireless networks for user authentication and key distribution.
- Mobile telephony has gone through different generations and multiple access technologies: 1G (FDMA), 2G (TDMA), 3G (CDMA), 4G (OFDM), and 5G (OFDM).
- Satellite communications links provide connectivity across very long distances and in places that would otherwise not be reachable, but may introduce latency challenges.

Questions

Please remember that these questions are formatted and asked in a certain way for a reason. Keep in mind that the CISSP exam is asking questions at a conceptual level. Questions may not always have the perfect answer, and the candidate is advised against always looking for the perfect answer. Instead, the candidate should look for the best answer in the list.

1. Which of the following is not a characteristic of the IEEE 802.11a standard?

A. It works in the 5-GHz range.

B. It uses the OFDM spread-spectrum technology.

C. It provides 52 Mbps in bandwidth.

D. It covers a smaller distance than 802.11b.

2. Wireless LAN technologies have gone through different versions over the years to address some of the inherent security issues within the original IEEE 802.11 standard. Which of the following provides the correct characteristics of WPA2 in Enterprise mode?

A. IEEE 802.1X, WEP, MAC

B. IEEE 802.1X, EAP, TKIP

C. IEEE 802.1X, EAP, WEP

D. IEEE 802.1X, EAP, CCMP

3. Which of the following is *not* a characteristic of Li-Fi networks?

A. Support for high client densities

B. High latency

C. Constrained coverage area

D. Can work on the infrared spectrum

4. How would you best ensure the security of a ZigBee system?

A. Ensure a coordinator acts as a Trust Center

B. Use 256-bit encryption keys

C. Deploy in a ring topology with preassigned slots for each device

D. Use the Symmetric-Key Key Establishment (SKKE) protocol to derive keys

5. Which of the following is a Bluetooth-specific attack that allows unauthorized read/write access from a wireless device?
   - **A.** Bluejacking
   - **B.** Replay attack
   - **C.** Smurf attack
   - **D.** Bluesnarfing
6. What does the IEEE 802.1X standard cover?
   - **A.** A Management Frame Protection (MFP) that prevents replay and denial-of-service (DoS) attacks
   - **B.** Wi-Fi Protected Access 2 (WPA2)
   - **C.** Security extensions to the physical layer (PHY) and Media Access Control (MAC) sublayer of the data link layer in the OSI model
   - **D.** An access control protocol for user authentication and key distribution
7. Which of the following is not a disadvantage of satellite networks compared to terrestrial ones?
   - **A.** Latency
   - **B.** Cost
   - **C.** Bandwidth
   - **D.** Video conferencing

Use the following scenario to answer Questions 8–10. You are planning an upgrade for the wireless network at one of your manufacturing sites and want to use this as an opportunity to improve network security. The current system is based on 10-year-old wireless access points (WAPs) that implement 802.11g. You're using WPA2 in Personal mode because you have multiple Industrial Internet of Things (IIoT) devices. You can update the firmware on the WAPs, but you really think it's time for an upgrade.

8. What could make it harder for you to switch from WPA2 Personal mode to Enterprise mode?
   - **A.** Enterprise mode requires licenses that can be costly.
   - **B.** The WAPs may not support Enterprise mode.
   - **C.** IIoT devices may not support Enterprise mode.
   - **D.** The return on investment is insufficient.
9. What is the best technology to which you should consider upgrading?
   - **A.** IEEE 802.16
   - **B.** IEEE 802.11w
   - **C.** IEEE 802.11f
   - **D.** IEEE 802.11ax

10. The existing wireless network has recently become unusable, and you suspect you may be the target of a persistent Wi-Fi deauthentication attack. How can you best mitigate this threat?
    - **A.** Deploy WPA3 access points across the facility
    - **B.** Perform MAC address filtering to keep the rogue stations off the network
    - **C.** Immediately update the firmware on the access points to support 802.11w
    - **D.** Change the channel used by the WAPs

Answers

1. **C.** The IEEE standard 802.11a uses the OFDM spread-spectrum technology, works in the 5-GHz frequency band, and provides bandwidth of up to 54 Mbps. The operating range is smaller because it works at a higher frequency.
2. **D.** Wi-Fi Protected Access 2 requires IEEE 802.1X or preshared keys for access control, Extensible Authentication Protocol (EAP) or preshared keys for authentication, and the Advanced Encryption Standard (AES) algorithm in counter mode with CBC-MAC Protocol (CCMP) for encryption.
3. **B.** Latency is the delay in data transfers, which is extremely low in Li-Fi networks.
4. **A.** Using a Trust Center provides a way to centrally authenticate devices and securely manage encryption keys, which are 128 bits (not 256). Without a Trust Center, the SKKE protocol can be used to derive keys, but this approach is not as secure. ZigBee does not support ring topologies.
5. **D.** Bluesnarfing could allow an attacker to read, modify, or delete calendar events, contacts, e-mails, text messages, and so on. Bluejacking is the only other Bluetooth attack option, but this refers to someone sending an unsolicited message to a device.
6. **D.** 802.1X is an access control protocol that can be implemented on both wired and wireless networks for user authentication and key distribution. MFP is covered in 802.11w, WPA2 is covered in 802.11i, and the other option (security extensions) was a distracter.
7. **C.** If you have the budget for it, data rates on satellite networks are comparable with other modes of communication. These systems, however, are typically more expensive and have high latencies, which means they are not well suited for time-sensitive applications, such as voice and video conferencing.
8. **D.** If a WAP supports WPA2, it would do so in either Personal or Enterprise mode as long as it can be connected to the needed backend services (e.g., a RADIUS server), with no need for additional licensing. Thus, the change would not typically be expected to have ROI issues. However, many embedded devices, including IIoT, do not support this mode and would have to be replaced.

9. **D.** 802.11ax is the only standard describing a WLAN among the list of options. 802.16 is used in metropolitan area networks (MANs). 802.11w covers Management Frame Protection (MFP) in wireless networks. 802.11f deals with users roaming among access points.

10. **C.** 802.11w provides Management Frame Protection (MFP) capabilities that would mitigate this type of attack. This is included in WPA3, so either answer would generally work. However, it is probably faster, cheaper, and safer to roll out 802.11w upgrades first, which would likely have no negative effects on the networks, while research and planning continue on how to best implement a WPA3 solution across the enterprise. This is a good example of the types of ambiguous questions you'll see on the CISSP exam.

This page intentionally left blank

CHAPTER 13

Securing the Network

This chapter presents the following:

- Secure networking
- Secure protocols
- Multilayer protocols
- Converged protocols
- Micro-segmentation

> *More connections to more devices means more vulnerabilities.*
>
> —Marc Goodman

Having developed a foundational understanding of networking technologies, we now turn our attention to building secure networks upon this foundation. In this chapter, we circle back to the core networking and service protocols introduced in Chapter 11 and discuss the threats against them and how to mitigate those threats. This discussion is grounded in the secure design principles covered in Chapter 9. We'll take the same approach as we expand our scope of interest from those core protocols and services to include other services, such as e-mail, that are critical to modern networks.

These networks are not as neatly divided as the OSI model could lead us to believe. Increasingly, we are relying on multilayer and converged protocols where concepts from different layers and even network components overlap in ways that have important security implications. The goal of this chapter is to show how, through a thoughtful application of secure protocols and best practices, we can secure our networks and the services they provide.

Applying Secure Design Principles to Network Architectures

A network architecture is just a model of a network. Like any model, it is not 100 percent representative of reality and uses abstractions to simplify some details so that we can focus on the others. By ignoring the little details (for now), we make it easier on ourselves to focus on the more important elements. For example, before we decide how many web servers we need and which operating systems and software we need to run on them, we should first identify the classes of servers and where we would put them. We might have a set of externally accessible servers for our web presence, but we may also need some servers that are for internal use only by all employees, and yet another set that is only for web developers. Where do we put each set and how might we need different controls for them? Maybe we need a demilitarized zone (DMZ), an internal sharing cluster, and a development virtual local area network (VLAN), each with specific sets of controls meant to mitigate their differing risk profiles. A network architecture allows us to answer these high-level questions before we start configuring any boxes.

Now, once we go through all the trouble of coming up with an architecture that works, we shouldn't have to reinvent the wheel. Network architectures also serve as templates for future systems. What's more, they can be codified and shared among similar organizations to reduce work and ensure we all follow best practices. Even if a lot of the details are different, a sound architecture can be reused time and again.

Many of these best practices relate to security. Since we intend our architectures to be reusable, it is imperative that we apply secure design principles when we implement them. In the sections that follow, we will discuss a (wide) variety of networking concepts and technologies that you will need to understand to implement secure design principles in network architectures. Periodically, we circle back and discuss some of these important secure design principles. It is important to note that there is no one-size-fits-all solution in this effort, so you will have to be selective about which of these principles you apply in any specific situation. Still, as a CISSP, you are expected to be conversant with all of them.

Let's start by reviewing the 11 secure design principles we covered in Chapter 9 and look at how they apply to network architectures.

- **Threat modeling** Everything we do in cybersecurity should be grounded in a good understanding of the threats we face. In this chapter, we focus our attention on network security, so we'll illustrate the threats we face as we discuss the various technologies and protocols involved in operating and securing our networks.
- **Least privilege** Traffic should be allowed to flow between any two points that are required to communicate in order to satisfy a valid organizational requirement, and nowhere else. We cover this in depth when we address network segmentation later in this chapter.
- **Defense in depth** While some IT and security professionals equate this principle with having a DMZ for public-facing servers, the principle applies throughout the network and requires that we build concentric defenses around our most valuable assets.
- **Secure defaults** Perhaps the simplest illustration of this principle as it applies to our networks is ensuring firewalls' default configurations are to deny all traffic from any source to any destination (deny all all). However, the principle should apply throughout our network and be consistent with least privilege.
- **Fail securely** The key to applying this principle is asking two questions: What happens when this network system fails? What happens when a packet doesn't match an "allow" rule on the firewall? (Hint: it should not be allowed through.)

- **Separation of duties** Speaking of firewall (and other security appliance) rules, who is in charge of those in your organization? Any sensitive duties should be split up among vetted staff members. At a minimum, if you don't have enough staff, everybody's sensitive work should be regularly checked by someone else.
- **Keep it simple** Unless you are architecting a global network for a multinational corporation, you should try to develop an architecture that can be depicted in a single PowerPoint slide and still describe all the important components.
- **Zero trust** Services and traffic on your network should all be authenticated and encrypted. When two servers are part of a system (e.g., the web server and its backend database), they should authenticate each other and have rules around what requests each is allowed to make of the other.
- **Privacy by design** Encrypting your network traffic is a good start toward protecting privacy, but where is the data being collected and for what purpose? For example, as we prepare for auditability (see the next principle), we need to ensure that we are not casting too wide of a net in terms of the data we log.
- **Trust but verify** Everything that happens on the network should be auditable, meaning that there should be a record of who is talking with whom, when, and why. This is normally done by ensuring logs are properly configured and protected against tampering or accidental loss.
- **Shared responsibility** Odds are that your network architecture will include at least a handful of service providers. Whether these are Internet service providers, cloud service providers, or managed services providers, it is critical to agree on who has responsibility over which aspects of your network.

EXAM TIP You should be prepared to map the various secure design principles to specific scenarios.

With these principles in mind, let's look at specific ways in which we can assess and implement network architectures securely.

Secure Networking

The most prevalent networking standards and protocols we use today (Ethernet, TCP/IP, and so on) were born decades ago (before many of us). Back then, the world was kinder and gentler (at least in the digital realm) and security just wasn't the sort of thing folks thought about when it came to computers and networks. With the explosion of the Internet came immense opportunities for both the law abiding and the criminals. The need for secure networking became apparent, but it was too late. We've been trying to catch up ever since by bolting security onto insecure technologies. One of the most common ways of securing our networks is through the use of encryption, particularly in trusted tunnels through untrusted networks.

Link Encryption vs. End-to-End Encryption

In each of the networking technologies discussed in this chapter, encryption can be performed at different levels, each with different types of protection and implications. Two general modes of encryption implementation are link encryption and end-to-end encryption. *Link encryption* encrypts all the data along a specific communication path, as in a satellite link, a terrestrial T3 leased line, or even between hosts on the same LAN. Because link encryption happens at layers 1 and 2, not only is the user information encrypted, but the (layer 3 and higher) headers, trailers, addresses, and routing data that are part of the packets are also encrypted. The only traffic not encrypted in this technology is the data link control messaging information, which includes instructions and parameters that the different link devices use to synchronize communication methods. Reading this information won't give an attacker any insights into what is being transmitted or where it is ultimately going.

End-to-end encryption (E2EE) occurs at the session layer (or higher), which means the headers, addresses, routing information, and trailer information are not encrypted, enabling attackers to learn more about a captured packet and where it is headed. Transport Layer Security (TLS), which we will discuss shortly, is the most common example of E2EE. Because the routing information is sent in plaintext, attackers can perform traffic analysis to learn details about the network, such as which hosts play which roles in it.

Link encryption, which is sometimes called *online encryption*, is usually provided by service providers and is incorporated into network protocols. All of the information is encrypted, and the packets must be decrypted at each hop so the router, or other intermediate device, knows where to send the packet next. The router must decrypt the header portion of the packet, read the routing and address information within the header, and then re-encrypt it and send it on its way.

With end-to-end encryption, the packets do not need to be decrypted and then encrypted again at each hop because the headers and trailers are not encrypted. The devices in between the origin and destination just read the necessary routing information and pass the packets on their way.

End-to-end encryption is usually initiated by the user of the originating computer. It provides more flexibility for the user to be able to determine whether or not certain

Encryption at Different Layers

Encryption can (and typically does) happen at different layers of an operating system and network stack. The following are just a few examples:

- End-to-end encryption happens within the applications.
- TLS encryption takes place at the session layer.
- Point-to-Point Tunneling Protocol (PPTP) encryption takes place at the data link layer.
- Link encryption takes place at the data link and physical layers.

messages will get encrypted. It is called "end-to-end encryption" because the message stays encrypted from one end of its journey to the other. Link encryption has to decrypt the packets at every device between the two ends.

Link encryption occurs at the data link and physical layers, as depicted in Figure 13-1. Hardware encryption devices interface with the physical layer and encrypt all data that passes through them. Because no part of the data is available to an attacker, the attacker cannot learn basic information about how data flows through the environment. This is referred to as *traffic-flow security*.

NOTE A *hop* is a device that helps a packet reach its destination. It is usually a router that looks at the packet address to determine where the packet needs to go next. Packets usually go through many hops between the sending and receiving computers.

Advantages of end-to-end encryption include the following:

- It provides more flexibility to the user in choosing what gets encrypted and how.
- Higher granularity of functionality is available because each application or user can choose specific configurations.
- Each hop device on the network does not need to have a key to decrypt each packet.

The disadvantage of end-to-end encryption is the following:

- Headers, addresses, and routing information are not encrypted, and therefore not protected.

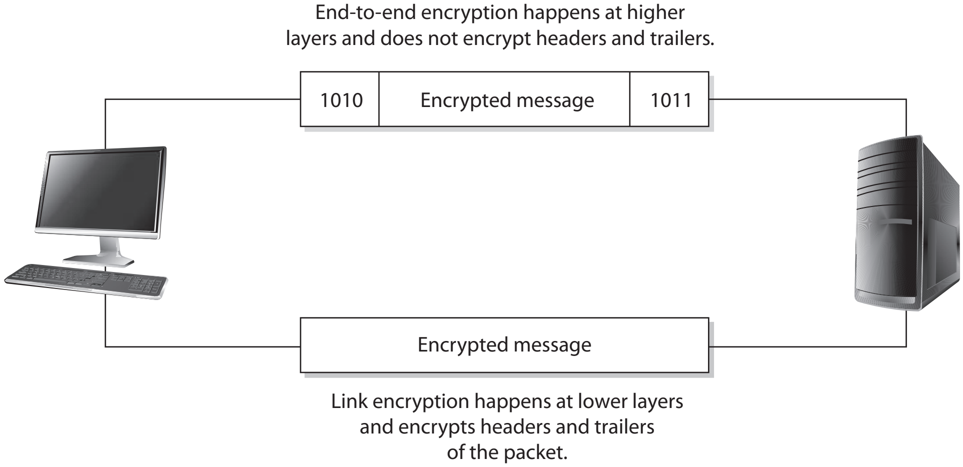

End-to-end encryption happens at higher layers and does not encrypt headers and trailers.
1010
Encrypted message
1011
Encrypted message
Link encryption happens at lower layers and encrypts headers and trailers of the packet.

Figure 13-1 Link and end-to-end encryption happen at different OSI layers.

Hardware vs. Software Cryptography Systems

Encryption can be done through software or hardware, and there are trade-offs with each. Generally, software is less expensive and provides a slower throughput than hardware mechanisms. Software cryptography methods can be more easily modified and disabled compared to hardware systems, but it depends on the application and the hardware product.

If an organization needs to perform high-end encryption functions at a higher speed, it will most likely implement a hardware solution.

Advantages of link encryption include the following:

- All data is encrypted, including headers, addresses, and routing information.
- Users do not need to do anything to initiate it. It works at a lower layer in the OSI model.

Disadvantages of link encryption include the following:

- Key distribution and management are more complex because each hop device must receive a key, and when the keys change, each must be updated.
- Packets are decrypted at each hop; thus, more points of vulnerability exist.

TLS

The most prevalent form of end-to-end encryption is *Transport Layer Security (TLS)*. TLS is a security protocol that provides confidentiality and data integrity for network communications. It replaced the (now insecure) Secure Sockets Layer (SSL) standard. These two protocols coexisted for many years, and most people thought that there were very few differences between SSL and TLS (TLS is currently in version 1.3). However, the Padding Oracle On Downgraded Legacy Encryption (POODLE) attack in 2014 was the death knell of SSL and demonstrated that TLS was superior security-wise. The key to the attack was to force SSL to downgrade its security, which was allowed for the sake of interoperability.

EXAM TIP Because SSL and TLS were (for a time) very closely related, the terms are sometimes still used interchangeably to describe network encryption in general. However, the SSL protocol has been insecure for many years and should not be the correct answer to an encryption question (unless it is asking for an insecure protocol).

Backward compatibility has long been a thorn in the side of those of us trying to improve cybersecurity. TLS 1.3 represents a switch to a focus on security, which shows in the limited number of cipher suites that it supports (just five). This means attackers can no longer trick a server into using an insecure cryptosystem during the connection establishment negotiation. One of the key features of TLS 1.3 is that the handshake used to establish a new connection requires only one client message to the server and one response from the server. There's a lot that happens in there, though, so let's take a look at a summarized version of this handshake.

1. Client "Hello" message, which includes
    - A list of cipher suites and protocols supported by the client
    - Client inputs for the key exchange
2. Server "Hello" message, which includes
    - The server's selection of cipher suite and protocol version
    - Server inputs for the key exchange
3. Server authentication, which includes
    - The server's digital certificate
    - Proof that the server owns the certificate's private key
4. (Optionally) Client authentication, which includes
    - The client's digital certificate
    - Proof that the client owns the certificate's private key

NOTE While TLS 1.3 minimizes the plaintext information transferred between hosts, TLS 1.2 (and earlier) passes a lot more information in the clear, potentially including the server name (e.g., www.goodsite.com).

As mentioned, TLS 1.3 has dramatically reduced the number of recommended cipher suites from 37 (in previous versions) to just five. This is an important improvement because some of those 37 suites were known (or suspected) to be vulnerable to cryptanalysis. By reducing the suites to five and ensuring these provide strong protection, TLS 1.3 makes it harder for attackers to downgrade the security of a system by forcing a server to use a weaker suite. The allowed suites in the latest version of TLS are as follows:

- **TLS_AES_256_GCM_SHA384** The encryption algorithm here is AES with a 256-bit key in Galois/Counter Mode (GCM). GCM is a mode of operation that provides message authentication. The hashing algorithm is SHA-384. This suite provides the best protection but requires the most computing resources.
- **TLS_AES_128_GCM_SHA256** This suite is almost identical to the preceding one, but saves on resources by using a smaller 128-bit key for encryption and a slightly faster SHA-256 for hashing. It is ideally suited for systems with hardware support for encryption.
- **TLS_AES_128_CCM_SHA256** In this suite, AES (again, with a 128-bit key) runs in Counter mode with CBC-MAC (CCM), which uses 16-byte tags to provide message authentication (much like GCM does).

- **TLS_AES_128_CCM_8_SHA256** This suite is almost identical to the preceding one, but Counter mode with CBC-MAC uses 8-byte tags (instead of 16-byte ones), which makes it better suited for embedded devices.
- **TLS_CHACHA20_POLY1305_SHA256** The ChaCha stream cipher (doing 20 rounds), combined with the Poly1305 message authentication code (MAC), is a cipher suite that is a good choice for software-based encryption systems. Many modern systems rely on hardware-based encryption, so the authors of TLS 1.3 wanted to ensure the recommended suites supported multiple devices. Besides, it just makes sense to have at least one encryption algorithm that is not AES.

We already discussed AES (and briefly mentioned ChaCha20) in Chapter 8, and CCM in Chapter 12, but this is the first time we bring up GCM and Poly1305. These are approaches to provide authenticated symmetric key encryption. *Authenticated encryption (AE)* provides assurances that a message was not modified in transit and could only come from a sender who knows the secret key. This is similar to the MAC approach discussed in Chapter 8 but is applied to stream ciphers. TLS 1.3 takes the AE concept to the next level in what is known as *authenticated encryption with additional data (AEAD)*. AEAD essentially computes the MAC over both ciphertext and plaintext when these are sent together. For example, when sending network traffic, there are certain fields (e.g., source and destination addresses) that cannot be encrypted. An attacker could replay an encrypted message later using a different packet, but if we're using AEAD (as TLS 1.3 requires), this bogus packet would automatically be discarded.

Another key feature of TLS 1.3 (which was optional in TLS 1.2 and prior) is its use of *ephemeral keys*, which are only used for one communication session and then discarded, using the Diffie-Hellman Ephemeral (DHE) algorithm. This provides *forward secrecy* (sometimes called *perfect forward secrecy*), which means that if attackers were somehow able to crack or otherwise obtain the secret key, it would only give them the ability to decrypt a small portion of the ciphertext. They wouldn't be able to decrypt everything going forward.

Attackers Use TLS Too!

While TLS is often our first line of defense in protecting our network traffic from prying eyes, attackers use it too, precisely for the same reason. There are many known examples of malware using TLS. Banking Trojans, such as TrickBot, Emotet, and Dyre, make use of TLS to communicate data back to their master server. Ransomware families, such as Jigsaw, Locky, and Petya, have also used TLS to infect machines and transfer information. They way in which attackers use TLS, however, is usually quite different from how it is used in legitimate connections. Analyzing network traffic can often point out some of these differences, such as:

- Offering weak or obsolete cipher suites
- Rarely offering more than one extension (enterprise clients use up to nine)
- Using self-signed certificates

While we focused on TLS 1.3 in this section, it is worth noting that, as of this writing, the Internet Society reports that only 58 percent of the world's top 1,000 websites support this latest version. What does this mean to you? You should balance the enhanced security of this protocol with the needs of your stakeholders. If you are not on TLS 1.3 yet, you may want to ask what percentage of your user base would not be able to communicate securely if you switched. All major browsers support it, so odds are that you'd be in good shape. But even if you're still on TLS 1.2, keep in mind that most of the features described in this section that make 1.3 so much better are optional in the previous version. This should give you a path to gradually improve your security while taking care of your stakeholders. Whatever your situation, TLS is probably the most important encryption technology for securing our networks, particularly our virtual private ones.

NOTE TLS 1.0 and TLS 1.1 were never formally deprecated but are widely considered insecure.

VPN

A *virtual private network (VPN)* is a secure, private connection through an untrusted network, as shown in Figure 13-2. It is a private connection because the encryption and tunneling protocols are used to ensure the confidentiality and integrity of the data in transit. It is important to remember that VPN technology requires a tunnel to work and it assumes encryption.

We need VPNs because we send so much confidential information from system to system and network to network. The information can be credentials, bank account data, Social Security numbers, medical information, or any other type of data we do not want to share with the world. The demand for securing data transfers has increased over the years, and as our networks have increased in complexity, so have our VPN solutions.

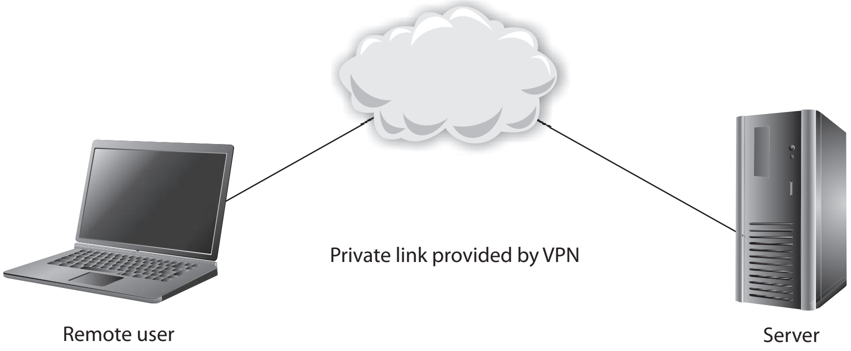


Figure 13-2 A VPN provides a virtual dedicated link between two entities across a public network.

Point-to-Point Tunneling Protocol

One of the early approaches to building VPNs was Microsoft's *Point-to-Point Tunneling Protocol (PPTP)*, which uses Generic Routing Encapsulation (GRE) and TCP to encapsulate Point-to-Point Protocol (PPP) connections and extend them through an IP network (running on TCP port 1723, by default). Since most Internet-based communication first started over telecommunication links, the industry needed a way to secure PPP connections, which were prevalent back then. The original goal of PPTP was to provide a way to tunnel PPP connections through an IP network, but most implementations included security features also since protection was becoming an important requirement for network transmissions at that time. PPTP, like many security protocols, did not age well and is now considered insecure and obsolete.

Layer 2 Tunneling Protocol

The *Layer 2 Tunneling Protocol (L2TP)*, currently in version 3, is a combination of Cisco's *Layer 2 Forwarding (L2F)* protocol and Microsoft's PPTP. L2TP tunnels PPP traffic over various network types (IP, ATM, X.25, etc.); thus, it is not just restricted to IP networks as PPTP was. PPTP and L2TP have very similar focuses, which is to get PPP traffic to an end point that is connected to some type of network that does not understand PPP. Unlike PPTP, L2TP runs on UDP (default port 1701), which makes it a bit more efficient. However, just like PPTP, L2TP does not actually provide much protection for the PPP traffic it is moving around, but it integrates with protocols that *do* provide security features. L2TP inherits PPP authentication and integrates with IPSec to provide confidentiality, integrity, and potentially another layer of authentication.

It can get confusing when several protocols are involved with various levels of encapsulation, but if you do not understand how they work together, you cannot identify if certain traffic links lack security. To figure out if you understand how these protocols work together and why, ask yourself these questions:

1. If the Internet is an IP-based network, why do we even need PPP?
2. If L2TP does not actually secure data, then why does it even exist?
3. If a connection is using IP, PPP, and L2TP, where does IPSec come into play?

Let's go through the answers together. Let's say that you are a remote user and work from your home office. You do not have a dedicated link from your house to your company's network; instead, your traffic needs to go through the Internet to be able to communicate with the corporate network. The line between your house and your ISP is a point-to-point telecommunications link, one point being your home router and the other point being the ISP's switch, as shown in Figure 13-3. Point-to-point telecommunication devices do not understand IP, so your router has to encapsulate your traffic in a protocol the ISP's device will understand—PPP. Now your traffic is not headed toward some website on the Internet; instead, it has a target of your company's corporate network. This means that your traffic has to be "carried through" the Internet to its ultimate destination through a tunnel. The Internet does not understand PPP, so your PPP traffic has to be encapsulated with a protocol that can work on the Internet and create the needed tunnel.

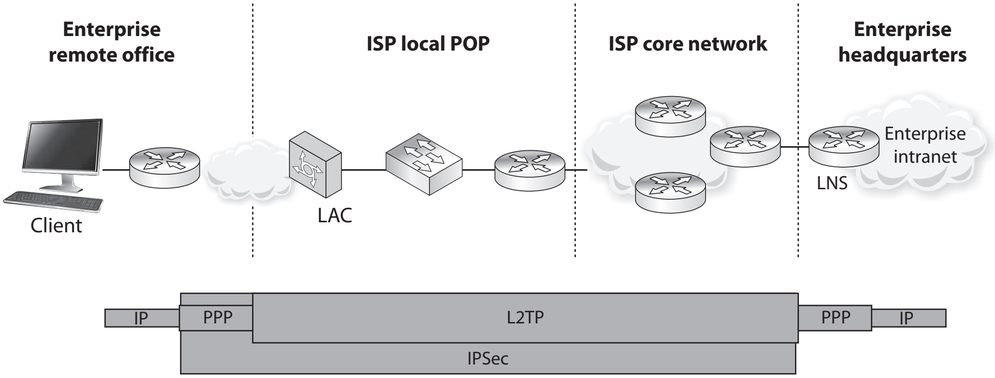


Figure 13-3 IP, PPP, L2TP, and IPSec can work together.

So your IP packets are wrapped up in PPP, and are then wrapped up in L2TP. But you still have no encryption involved, so your data is actually not protected. This is where IPSec comes in. IPSec is used to encrypt the data that will pass through the L2TP tunnel. Once your traffic gets to the corporate network's perimeter device, it will decrypt the packets, take off the L2TP and PPP headers, add the necessary Ethernet headers, and send these packets to their ultimate destination.

Here are the answers to our questions:

1. If the Internet is an IP-based network, why do we even need PPP?
   Answer: The point-to-point line devices that connect individual systems to the Internet do not understand IP, so the traffic that travels over these links has to be encapsulated in PPP.
2. If L2TP does not actually secure data, then why does it even exist?
   Answer: It extends PPP connections by providing a tunnel through networks that do not understand PPP.
3. If a connection is using IP, PPP, and L2TP, where does IPSec come into play?
   Answer: IPSec provides the encryption, data integrity, and system-based authentication.

Here is another question: Does all of this PPP, L2TP, and IPSec encapsulation have to happen for every single VPN used on the Internet? No, only when connections over point-to-point connections are involved. When two gateway routers are connected over the Internet and provide VPN functionality, they only have to use IPSec.

Internet Protocol Security

Internet Protocol Security (IPSec) is a suite of protocols that was developed to specifically protect IP traffic. IPv4 does not have any integrated security, so IPSec was developed to "bolt onto" IP and secure the data the protocol transmits. Where L2TP works at the data link layer, IPSec works at the network layer of the OSI model.

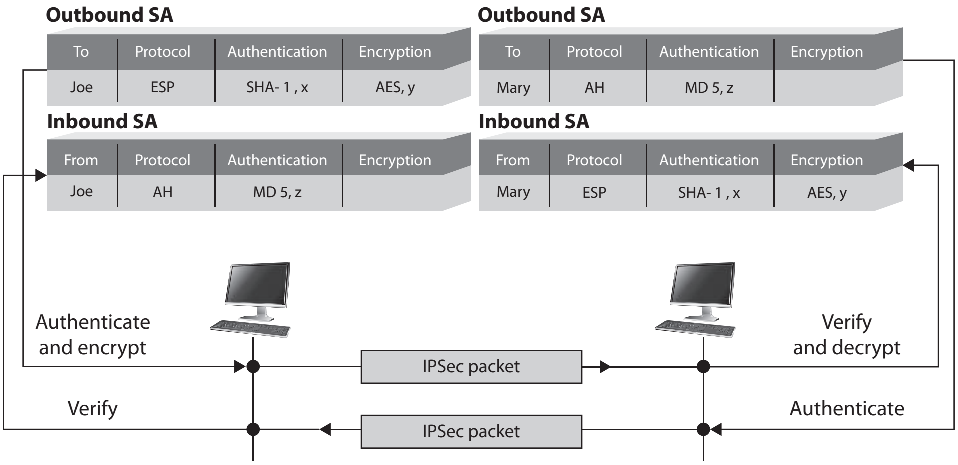
The main protocols that make up the IPSec suite and their basic functionality are as follows:

- **Authentication Header (AH)** Provides data integrity, data-origin authentication, and protection from replay attacks
- **Encapsulating Security Payload (ESP)** Provides confidentiality, data-origin authentication, and data integrity
- **Internet Security Association and Key Management Protocol (ISAKMP)** Provides a framework for security association creation and key exchange
- **Internet Key Exchange (IKE)** Provides authenticated keying material for use with ISAKMP

AH and ESP can be used separately or together in an IPSec VPN configuration. The AH protocols can provide data-origin authentication (system authentication) and protection from unauthorized modification, but do not provide encryption capabilities. If the VPN needs to provide confidentiality, then ESP has to be enabled and configured properly.

When two routers need to set up an IPSec VPN connection, they have a list of security attributes that need to be agreed upon through handshaking processes. The two routers have to agree upon algorithms, keying material, protocol types, and modes of use, which will all be used to protect the data that is transmitted between them.

Let's say that you and Juan are routers that need to protect the data you will pass back and forth to each other. Juan sends you a list of items that you will use to process the packets he sends to you. His list contains AES-128, SHA-1, and ESP tunnel mode. You take these parameters and store them in a security association (SA). When Juan sends you packets one hour later, you will go to this SA and follow these parameters so that you know how to process this traffic. You know what algorithm to use to verify the integrity of the packets, the algorithm to use to decrypt the packets, and which protocol to activate and in what mode. Figure 13-4 illustrates how SAs are used for inbound and outbound traffic.



Outbound SA

To	Protocol	Authentication	Encryption
Joe	ESP	SHA- 1 , x	AES, y

Inbound SA

From	Protocol	Authentication	Encryption
Joe	AH	MD 5, z	

Outbound SA

To	Protocol	Authentication	Encryption
Mary	AH	MD 5, z	

Inbound SA

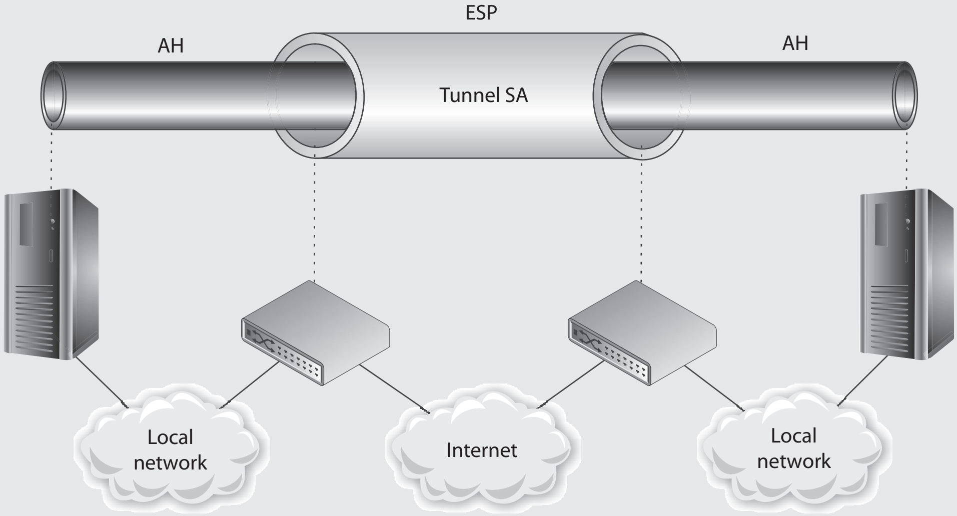
From	Protocol	Authentication	Encryption
Mary	ESP	SHA- 1 , x	AES, y

Figure 13-4 IPSec uses security associations to store VPN parameters.

NOTE The U.S. National Security Agency (NSA) uses a protocol encryptor that is based upon IPSec. A *HAIPE (High Assurance Internet Protocol Encryptor)* is a Type 1 encryption device that is based on IPSec with additional restrictions, enhancements, and capabilities. A HAIPE is typically a secure gateway that allows two enclaves to exchange data over an untrusted or lower-classification network. Since this technology works at the network layer, secure end-to-end connectivity can take place in heterogeneous environments. This technology has largely replaced link layer encryption technology implementations.

IPSec

IPSec can be configured to provide *transport adjacency*, which just means that more than one security protocol (ESP and AH) is used in a VPN tunnel. IPSec can also be configured to provide *iterated tunneling*, in which an IPSec tunnel is tunneled through another IPSec tunnel, as shown in the following diagram. Iterated tunneling would be used if the traffic needed different levels of protection at different junctions of its path. For example, if the IPSec tunnel started from an internal host and terminated at an internal border router, this may not require encryption, so only the AH protocol would be used. But when that data travels from that border router throughout the Internet to another network, then the data requires more protection. So the first packets travel through a semisecure tunnel until they get ready to hit the Internet and then they go through a very secure second tunnel.



The most common implementation types of TLS VPN are as follows:

- **TLS portal VPN** An individual uses a single standard TLS connection to a website to securely access multiple network services. The website accessed is typically called a *portal* because it is a single location that provides access to other resources. The remote user accesses the TLS VPN gateway using a web browser, is authenticated, and is then presented with a web page that acts as the portal to the other services.
- **TLS tunnel VPN** An individual uses a web browser to securely access multiple network services, including applications and protocols that are not web-based, through a TLS tunnel. This commonly requires custom programming to allow the services to be accessible through a web-based connection.

Summary of Tunneling Protocols

Layer 2 Tunneling Protocol (L2TP):

- Hybrid of L2F and PPTP
- Extends and protects PPP connections
- Works at the data link layer
- Transmits over multiple types of networks, not just IP
- Combined with IPSec for security

IPSec:

- Handles multiple VPN connections at the same time
- Provides secure authentication and encryption
- Supports only IP networks
- Focuses on LAN-to-LAN communication rather than user-to-user communication
- Works at the network layer and provides security on top of IP

Transport Layer Security (TLS):

- Works at the session layer and protects mainly web and e-mail traffic
- Offers granular access control and configuration
- Easy to deploy since TLS is already embedded into web browsers
- Can only protect a small number of protocol types, thus is not an infrastructure-level VPN solution

Since TLS VPNs are closer to the application layer, they can provide more granular access control and security features compared to the other VPN solutions. But since they are dependent on the application layer protocol, there are a smaller number of traffic types that can be protected through this VPN type.

One VPN solution is not necessarily better than the other; they just have their own focused purposes:

- L2TP is used when a PPP connection needs to be extended through a network.
- IPSec is used to protect IP-based traffic and is commonly used in gateway-to-gateway connections.
- TLS VPN is used when a specific application layer traffic type needs protection.

Secure Protocols

TLS may be one of the most talked-about technologies when it comes to network security. Still, there are other protocols, and other applications of TLS, that you should know. This section addresses each of the main network services, web, DNS, and e-mail. Let's start with how we secure web services.

Web Services

Many people hear the term "web services" and think of websites and the web servers that do the work behind the scenes. In reality, however, this is but a portion of what the term actually covers. A *web service* is a client/server system in which clients and servers communicate using HTTP over a network such as the Internet. Sure, this definition covers static web pages written in HTML being served out of an old Apache server somewhere, but it can also cover much more.

For example, suppose you are a retailer and don't want to pay for a huge storage space for merchandise that may or may not sell anytime soon. You could implement a just-in-time logistics system that keeps track of your inventory and past selling patterns, and then automatically order merchandise so that it arrives just before you start running low. This kind of system is typically implemented using a business-to-business (B2B) web service and is depicted in Figure 13-5. Each icon in the figure represents a distinct web service component.

When we look at web services this way, it becomes clear that we have much more to worry about than simply the interaction between customers and our website. Let's look at ways in which we could implement some of the secure design principles in this example. The following list is meant to be illustrative, not all-inclusive:

- **Least privilege** The forecasting service should have read-only access to some of the data in the inventory system. It doesn't need any additional access.
- **Secure defaults** The inventory service should refuse all connection requests from any endpoint other than those explicitly authorized (point of sale and forecasting). If any other connections are required, those should be added as exceptions after a careful review.

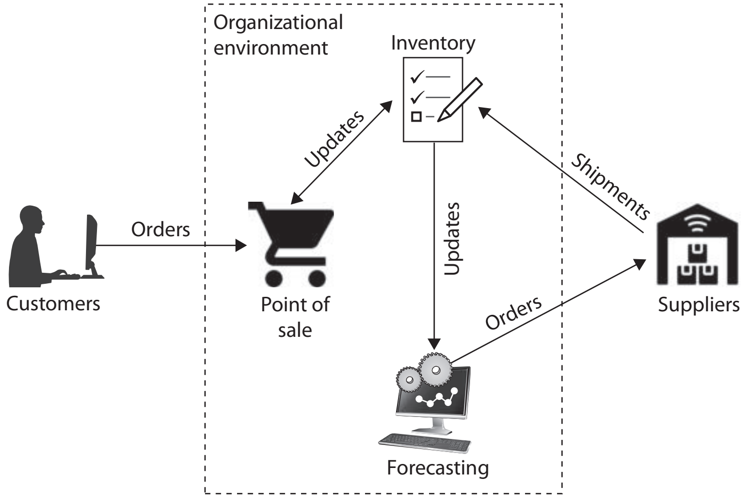


Figure 13-5 Example just-in-time logistics B2B web service

- **Fail securely** The forecasting service has the ability to spend money by placing orders from suppliers. It should not process any order that is malformed or otherwise fails any checks.
- **Separation of duties** The forecasting service can place orders but cannot receive shipments and update inventory. Ordering and receiving should be two separate duties performed by different people/systems to mitigate the risk of fraud.
- **Zero trust** Before any two components collaborate, they should both be required to authenticate with each other and encrypt all communications. This is particularly true (and the authentication protocol should be more rigorous) when dealing with external parties like customers and suppliers.
- **Privacy by design** Customer information should not be shared outside the point-of-sale (PoS) system, particularly since the other two internal systems (inventory and forecasting) communicate with an external third party. This example is overly simplistic, but the point is that customer data should be limited to the components that absolutely need it.
- **Trust but verify** All components (with the possible exception of the user) should generate logs that are sufficient to detect attacks or errors. Ideally, these logs are centrally collected to make them easier to correlate and harder to tamper with.
- **Shared responsibility** The security obligations of the organization and of the supplier should be codified in a legally binding contract and audited periodically.

Again, the list is not exhaustive, but it should give you an idea of how the secure design principles can be applied to a web services scenario. You should be prepared to do likewise with a variety of other scenarios for the CISSP exam.

How are these web services actually delivered? The key is to focus on *what* service is being delivered, and not on *how* it is implemented or *where* it is hosted (as long as it is available). A *service-oriented architecture (SOA)* describes a system as a set of interconnected but self-contained components that communicate with each other and with their clients through standardized protocols. These protocols, called *application programming interfaces (APIs)*, establish a "language" that enables a component to make a request from another component and then interpret that second component's response. The requests that are defined by these APIs correspond to discrete business functions (such as estimated shipping costs to a postal code) that can be useful by themselves or can be assembled into more complex business processes. An SOA has three key characteristics: self-contained components, a standardized protocol (API) for requests/responses, and components that implement business functions.

SOAs are commonly built using web services standards that rely on HTTP as a standard communication protocol. Examples of these are SOAP (which used to stand for the Simple Object Access Protocol) and the Representational State Transfer (REST) architectures. Let's look at these three (HTTP, SOAP and REST) in turn.

Hypertext Transfer Protocol

HTTP is a TCP/IP-based communications protocol used for transferring resources (e.g., HTML files and images) between a server and a client. It also allows clients to send queries to the server. The two basic features of HTTP are that it is connectionless and stateless. Connectionless protocols do not set up a connection (obviously) and instead send their messages in a best-effort manner. They rely on some other protocol (in this case TCP) to ensure the message gets across. *Stateless* means that the server is amnesiac; it doesn't remember any previous conversations with any clients. Thus, whatever is needed for the server to "remember" has to be provided with each request. This is a role commonly played by session identifiers and cookies.

NOTE A cookie is just a small text file containing information that only one website can write or read.

Uniform Resource Identifiers A foundational component of HTTP is the use of the uniform resource identifier (URI), which uniquely identifies a resource on the Internet. A typical URI looks like this: http://www.goodsite.com:8080/us/en/resources/search.php?term=cissp. Let's look at its components in sequence:

1. **Scheme** This is another name for the protocol being used (e.g., HTTP or HTTPS) and ends in a colon (:).
2. **Authority** There are three possible subcomponents here, but the second is the most prevalent:
    - Username (optional) (and optional password, separated by a colon) followed by an at (@) symbol.
    - Host in either hostname (e.g., www.goodsite.com) or IP address format.
    - Port number (optional), preceded by a colon (e.g., :8080). Note that port 80 is assumed for HTTP schemes and port 443 for HTTPS schemes.

3. **Path** The path to the requested resource on the server. If the path is not specified by the client, it is assumed to be a single slash (/), which is the default document at the root of the website (e.g., the homepage). Subdirectories are indicated as they are in Linux/Unix by successive slashes (e.g., /us/en/resources/search.php).
4. **Query (optional)** An attribute-value pair preceded by a question mark (?) (e.g., ?term=cissp). Each additional pair is separated from the previous one by an ampersand (&).

Request Methods HTTP uses a request-response model in which the client requests one or more resources from the server, and the latter provides the requested resources (assuming, of course, they are available to the client). The protocol defines two request methods: GET and POST. The main difference for our discussion is that a GET request must include all parameters in the URI, while POST allows us to include additional information (e.g., parameters) in the body of the request, where it will not be revealed in the URI. So, in the previous example we can safely guess that the method used was GET because we see the search term (cissp) in the URI.

Hypertext Transfer Protocol Secure *HTTP Secure (HTTPS)* is HTTP running over Transport Layer Security (TLS). Ensuring that all your web services require HTTPS is probably the most important security control you can apply to them. Recall that unencrypted requests can provide an awful lot of sensitive data, including credentials, session IDs, and URIs. Ideally, you require TLS 1.3 on all your web servers and ensure they do not allow unencrypted communications (by enforcing secure defaults).

An important consideration before you jump to HTTPS everywhere is whether you want to perform deep packet analysis on all your internal traffic. If you force use of HTTPS, you will need to deploy TLS decryption proxies, which can be pricey and require careful configuration on all your endpoints. The way these proxies work is by performing what is essentially a (benign) man-in-the-middle attack in which they terminate the clients' secure sessions and establish the follow-on session to their intended server. This allows the proxy to monitor all HTTPS traffic, which provides a measure of defense in depth but may pose some challenges to the privacy by design principle. Many organizations deal with this challenge by whitelisting connections to certain types of servers (e.g., healthcare and financial services organizations), while intercepting all others.

SOAP

SOAP is a messaging protocol that uses XML over HTTP to enable clients to invoke processes on a remote host in a platform-agnostic way. SOAP was one of the first SOAs to become widely adopted. SOAP consists of three main components:

- A message envelope that defines the messages that are allowed and how they are to be processed by the recipient
- A set of encoding rules used to define data types
- Conventions regarding what remote procedures can be called and how to interpret their responses

Extensible Markup Language

The term XML keeps coming up for good reasons. Extensible Markup Language is a popular language to use if you want to mark up parts of a text document. If you've ever looked at raw HTML documents, you probably noticed the use of tags such as <title>CISSP</title> to mark up the beginning and end of a page's title. These tags enable both humans and machines to interpret text and process it (such as rendering it in a web browser) as the author intended. Similarly, XML enables the author of a text document to "explain" to a receiving computer what each part of the file means so that a receiving process knows what to do with it. Before XML, there was no standard way to do this, but nowadays there are a number of options, including JavaScript Object Notation (JSON) and YAML Ain't Markup Language (YAML).

SOAP security is enabled by a set of protocol extensions called the Web Services Security (WS-Security or WSS) specification, which provides message confidentiality, integrity, and authentication. Note that, in keeping with HTTP's stateless nature, the focus here is on message-level security. Confidentiality is provided through XML encryption, integrity through XML digital signatures, and single-message authentication through security tokens. These tokens can take on various forms (the specification is intentionally broad here), which include username tokens, X.509 digital certificates, SAML assertions, and Kerberos tickets (we'll cover the last two in Chapter 17).

One of the key features of SOAP is that the message envelope allows the requester to describe the actions that it expects from the various nodes that respond. This feature supports options such as routing tables that specify the sequence and manner in which a series of SOAP nodes will take action on a given message. This can make it possible to finely control access as well as efficiently recover from failures along the way. This richness of features, however, comes at a cost: SOAP is not as simple as its name implies. In fact, SOAP systems tend to be fairly complex and cumbersome, which is why many web service developers prefer more lightweight options like REST.

Representational State Transfer

Unlike SOAP, which is a messaging protocol, Representational State Transfer (REST) is an architectural pattern used to develop web services using a variety of languages. In REST, HTTP is used to provide an API that allows clients to make programmatic requests from servers. For example, a client of a RESTful service could insert a new user record using the HTTP POST method (which lets you send additional information in the body of the request) by sending the following URI: https://www.goodsite.com/UserService/Add/1. The server would know to read the body of the POST to get the new user's details, create it, and then send a HTTP confirmation (or error). As you can see, REST essentially creates a programming language in which every statement is an HTTP URI.

Because every interaction with the system is spelled out in the URI, it is essential to use HTTPS as a secure default communications protocol. Of course, in keeping with the principle of zero trust, we want to authenticate clients and servers to each other, as well as put limits on what resources are available to each client. Another good security practice for RESTful services, which applies to any software system, is to validate all inputs before processing them. This mitigates a large number of possible injection attacks in which the adversary deliberately provides malformed inputs in order to trigger a system flaw.

Domain Name System

We covered the Domain Name System (DNS) in a fair amount of detail back in Chapter 11. Let's return to it now in the context of its role in helping us to secure our networks. Early on in its history, DNS was most commonly targeted by attackers to hijack requests, redirecting the unwitting requesters to malicious hosts instead of the legitimate ones they were seeking. While this is still a concern that we'll address in a bit, we also have to consider the much more common use of DNS to assist threat actors in conducting attacks, rather than being the target of attacks.

Since some of the most problematic adversarial uses of DNS depend on how this system works, let's review the process by which DNS performs recursive queries. Recall from Chapter 11 that a *recursive query* means that the request can be passed on from one DNS server to another one until the DNS server with the correct information is identified. This is illustrated in Figure 13-6. First, the client queries its local DNS server, which may either be an authoritative source for it or have cached it after some other client's request. Failing that, the server will typically start by consulting the root DNS server. The root server (there are actually a few of them for redundancy) will probably say something like "No, but here is the address of the name server for all .com domains." The local server will then query that server, which will probably result in it responding "No, but here is the address of the name server responsible for ironnet.com." Finally, the local server will query that other server, which will respond with an A record containing the IP address of the www host.

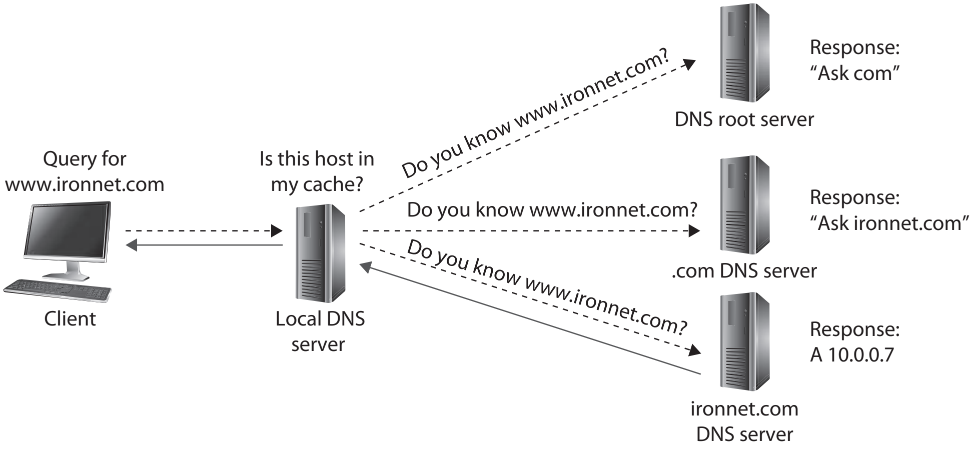


Figure 13-6 A recursive DNS query

Preventing Common DNS Attacks

DNS is the Internet's ubiquitous messenger; its queries and responses go everywhere, and without them the Internet as we know it would not work. Because of its importance to most other network systems, DNS traffic is seldom blocked by firewalls or routers. Attackers quickly figured out that this ubiquity makes DNS a preferred tool to manipulate and use for their own nefarious purposes. Perhaps the cleverest application of DNS for unintended purposes is its use to reach out and touch hosts in ways that are difficult to block using pseudo-randomly generated domain names.

EXAM TIP You will not be tested on the material that covers the following DNS attacks, but note that these attacks are both important to know and illustrative of the challenges we face in securing networks. If you are preparing for the exam only, feel free to move to the "Domain Name System Security Extensions" section.

Domain Generation Algorithms Once malware is implanted on target systems, the adversaries still need to communicate with those hosts. Since inbound connection attempts would easily be blocked at the firewall, most malware initiates outbound connections to the attacker's command and control (C2) infrastructure instead. The problem for the attackers is that if they provide a hostname or IP address in the malware, defenders will eventually find it, share it as an indicator of compromise (IOC), and reduce or negate the effectiveness of the C2 system.

To bypass signature detection by intrusion detection systems (IDSs) and intrusion prevention systems (IPSs) that use these IOCs, malware authors developed algorithms that can generate different domain names in a manner that appears random but produces a predictable sequence of domain names for those who know the algorithm. Suppose I am an attacker and want to hide my real C2 domains to keep them from being blocked or removed. I develop a domain generation algorithm (DGA) that produces a new (seemingly) random domain name each time it is run. Sprinkled somewhere in that (very long) list of domains are the ones I actually want to use. The infected host then attempts to resolve each domain to its corresponding IP address using DNS. Most of the domains do not exist and others may be benign, so either way there is no malicious C2 communications that follow. However, since I know the sequence of domains generated by the DGA and I know how quickly the malware will generate them, I can determine approximately when a particular infected host will query a specific domain. I can then register it the day before and rendezvous with the malware on that domain so I can receive its report and/or issue commands. The defenders won't know which domains are my malicious ones and which are just noise meant to distract them.

Figure 13-7 shows three domains being generated by an infected host. The first two that are queried do not exist, and thus result in an NXDOMAIN response from the server, which means the domain was not found. The third domain resolves to a malicious domain. When the authoritative (malicious) server for that domain receives the request, it knows it comes from a compromised system and sends a response that, when decoded, means "sleep for 7 hours."

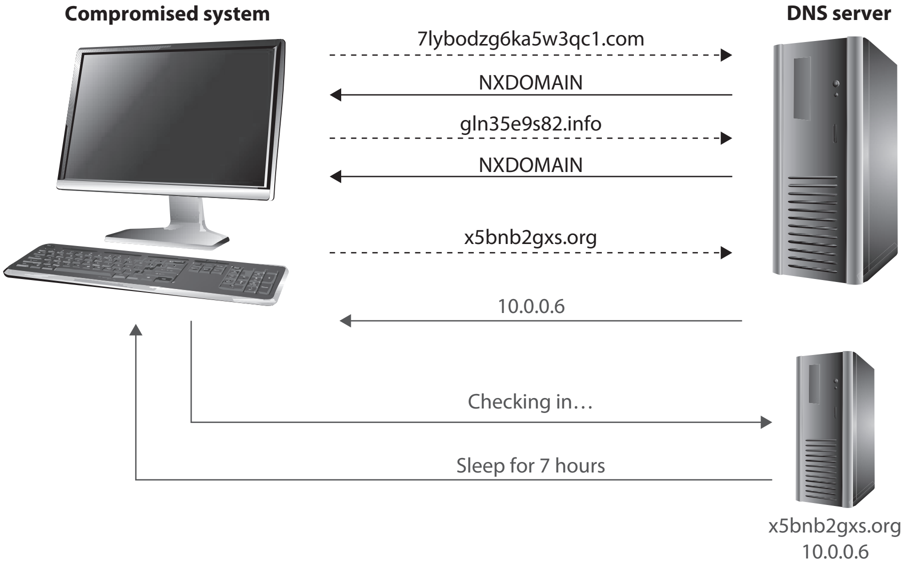


Figure 13-7 DGA in use by a compromised system

How can we detect and stop this kind of adversarial behavior? There are two general approaches. The first is to capture the malware and reverse engineer its DGA. We then play it forward (just like the attacker does) to determine which domains will be generated and when. Knowing this timeline, you can blacklist the domains and use the fact that a host attempted to reach them to infer that the querying system is compromised. Keep in mind that different compromised systems will be generating domain names at different times, so the task is onerous even for organizations that are mature enough to reverse engineer malware in the first place.

The second approach to detecting and stopping the use of DGAs is to analyze the domain names in each query to determine the probability of the query being legitimate. You can see from Figure 13-7 that most domains generated by these algorithms look, well, random. They are not the sort of domain names that you would expect someone to pay money to register. If you find a domain that is highly suspicious, you can investigate the host to see if it is infected, or you could block or monitor the DNS query and response to see if there is anything suspicious in either. For example, in some cases, the response will come as an encoded or encrypted message in a TXT record. This approach is only practical if you have a fairly sophisticated artificial intelligence analysis system that can examine every DNS request and learn over time which ones are likely to be bad.

NOTE There are legitimate uses of DGAs. For example, some systems use them to test whether or not a system can reach the Internet and perhaps track who that system is. This is done by some developers for licensing, updating, or diagnostic purposes.

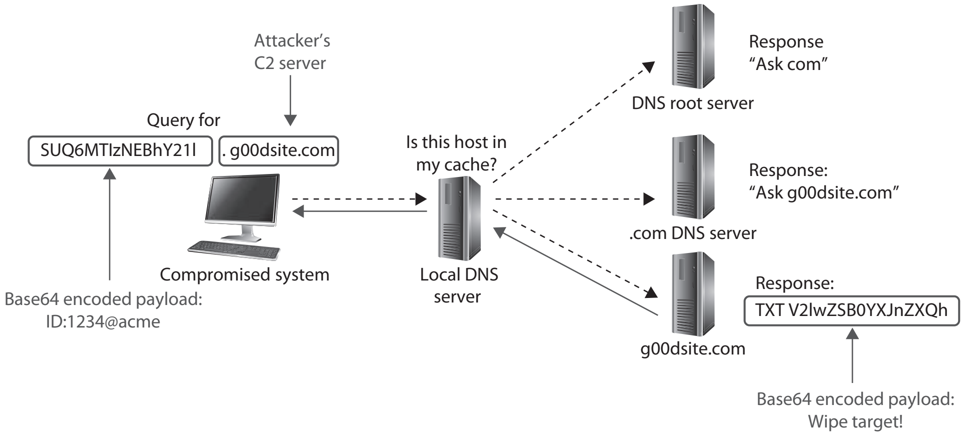


Figure 13-8 Covert communication over a DNS tunnel

DNS Tunneling Malicious use of a DGA can be very hard to stop unless you have advanced capabilities at your disposal. Fortunately, however, this use is limited to simple messaging between a compromised host and an external threat actor. But what if we could use DNS to transfer more information? A lot more? It turns out that data can be hidden in DNS queries using encoded host and other resource labels. *DNS tunneling* is the practice of encoding messages in one or a series of DNS queries or responses for exfiltrating or infiltrating data into an environment.

Figure 13-8 shows a very simple example of DNS tunneling that builds on our discussion of recursive queries in Figure 13-6. In this case, the compromised system wants to check in with its C2 server, so it uses Base64 encoding to obfuscate its message, which contains its identifier. Let's say that this is an infected host in the Acme Corporation, so its ID is 1234@acme. The recursive DNS query eventually is sent to the server that owns the malicious domain g00dsite.com. It decodes the hostname field, sees which of its bots this is from, and decides it is time to wipe the file system on the infected system. This command comes in the form of a TXT response that is also Base64 encoded.

A similar, but much less noticeable, use of DNS tunneling is to slowly exfiltrate data from the compromised system. Since DNS allows names of up to 63 characters between each dot, attackers can break down a longer file (e.g., a secret document) and exfiltrate it in a sequence of DNS queries to the same server or different servers.

Defending against DNS tunneling is similarly difficult to countering DGAs. Again, we could use network detection and response (NDR) solutions that use artificial intelligence to look for this type of behavior. However, because this type of attack (unlike DGAs) tends to rely on just a few domains, we could use domain reputation tools to determine whether any of our systems are making queries for suspicious or malicious domains.

Distributed Denial of Service The third type of DNS attack targets someone else's infrastructure using your DNS servers. An attacker who owns (or can rent) a large army of compromised systems (bots) can use them to overwhelm a target with name resolution responses to queries it didn't send out in the first place. To see how this attack works, we must first consider that DNS is based on UDP, which means spoofing the source address of a query is trivial.

In a *DNS reflection attack*, the threat actor instructs each bot they control to send a query to one of many open DNS servers around the world, while spoofing the source addresses on those queries. Collectively, the responding servers then bombard the target with traffic. If you have a sufficient number of bots and servers doing this quickly enough, the results could take the target system offline. Even if the target is not a DNS server, it still has to process millions (or more) of UDP packets arriving each second, which can overwhelm the typical server. But what if we could amplify the effects?

A *DNS amplification attack* is characterized by small queries that result in very much larger responses. A typical query is about 30 bytes and its response is around 45 bytes on average. The following are three techniques that are used to turn this relatively equal ratio of query to response size by a factor of up to 50 times:

- **DNS ANY** DNS has a (deprecated in 2019, but still used) diagnostic feature that allows a client to request all the information a server has on a given domain name. By sending a query of type ANY, an attacker can cause the server to send all the records in that domain up to the maximum size of a DNS message, which is 512 bytes. Having a 30-byte query produce a 512-byte response is a 17× amplification.
- **EDNS(0)** There are several situations in which the 512-byte limit on DNS messages over UDP becomes problematic. In particular, it is not possible to implement DNS Security Extensions (DNSSEC) with this constraint. Therefore, the Internet Engineering Task Force (IETF) developed EDNS(0), the Extension Mechanisms for DNS, which allows for up to 4096-byte responses. Properly used by an attacker, this new maximum size represents a 136× amplification given a 30-byte query.
- **DNSSEC** One of the most practical ways to exploit the maximum size defined in EDNS(0) is, ironically, using DNSSEC. Going back to Figure 13-6, when the local DNS server requests the A record from the authoritative server for that domain (the bottom left one), it also requests the DNSSEC associated with the zone. This is done to ensure the identity of the authoritative server (and hence the response) but results in a significantly larger response (because it includes a digital signature). So, all an attacker needs to do is find open DNS servers that have DNSSEC enabled and direct the bots at them.

Domain Name System Security Extensions

DNSSEC is a set of standards IETF developed to protect DNS from a variety of attacks. Specifically, DNSSEC is focused on ensuring the integrity of DNS records, not their confidentiality or availability. In plain-old DNS, a client makes a recursive query that, eventually, is responded to by some server that claims to be authoritative and provides an IP address. As we discussed in Chapter 11, however, this led to impersonation attacks where unwitting clients were pointed to malicious hosts. In response to this threat, the IETF came up with DNSSEC.

DNSSEC works by grouping records in a DNS zone according to their name and type (e.g., A, NS, MAIL) into Resource Record Sets (RRSets) that are then digitally signed, with the resulting signature going into a resource record signature (RRSig) record. The corresponding public key is published in a DNSKey record. So, when we want to resolve a fully qualified domain name (FQDN) using DNSSEC, we first retrieve the RRSet containing the name, then we request the RRSig for that set, and finally we verify that the record has not been tampered with. While this approach prevents impersonation and cache poisoning attacks, it has, as we just saw, also opened the door to crippling amplification attacks.

DNS over HTTPS

While DNSSEC ensures the integrity of DNS data, it does nothing to protect the confidentiality or privacy of queries. Sure, you can be confident that the IP address you got back was the right one, but what if anyone on the network can now see that you went to a domain called embarrassingmedicalcondition.com? We know from our discussion of TLS 1.3 earlier in this chapter that this URL will not go out in plaintext over HTTPS (which, by the way, it will in TLS 1.2 and earlier), but it will still be visible before the TLS handshake when the DNS query goes out. This is particularly problematic when we are connected to public networks such as the Wi-Fi network at the local coffee shop.

DNS over HTTPS (DoH) is a (yet to be ratified) approach to protecting the privacy and confidentiality of DNS queries by sending them over HTTPS/TCP/IP instead of unsecured UDP/IP. As of this writing, DoH is available on most platforms, though it is an optional feature that has to be configured. Keep in mind, however, that DoH provides confidentiality but (unlike DNSSEC) not integrity protections. Also, DoH was conceived as a privacy mechanism when using public networks. If you think back to the DNS-enabled attacks we discussed earlier in this chapter (especially DGA and DNS tunneling), DoH would actually make these much harder to detect unless you have a TLS decryption proxy in place. This is one of the reasons why the U.S. NSA recommended in 2021 that DoH not use external resolvers in enterprise networks.

DNS Filtering

Our final topic on securing DNS is perhaps the most obvious. Instead of allowing any DNS request to go out of our organizational networks, what if we first filtered them to block known malicious (or otherwise disallowed) domains from being resolved in the first place? A DNS filter performs a similar role as a web proxy that blocks content that is inappropriate, except that it works on DNS instead of HTTP traffic. There are many commercial solutions that provide this functionality, but keep in mind they should be deployed as part of a broader, defense-in-depth approach to securing DNS.

Electronic Mail

Let's now shift our attention to the third major service (along with web and DNS services) that is required for virtually all major organizations: e-mail. Though it has lost some ground to other business communication platforms such as Slack, Microsoft Teams,

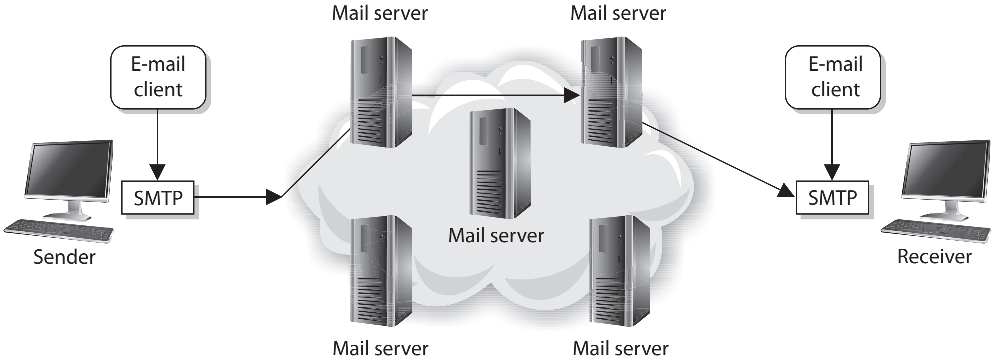


Figure 13-9 SMTP works as a transfer agent for e-mail messages.

and Google Hangouts, e-mail remains a critical service in virtually all organizations. An e-mail message, however, is of no use unless it can actually be sent somewhere. This is where *Simple Mail Transfer Protocol (SMTP)* comes in. In e-mail clients, SMTP works as a message transfer agent, as shown in Figure 13-9, and moves the message from the user's computer to the mail server when the user clicks the Send button. SMTP also functions as a message transfer protocol between e-mail servers. Lastly, SMTP is a message-exchange addressing standard, and most people are used to seeing its familiar addressing scheme: something@somewhere.com.

Many times, a message needs to travel throughout the Internet and through different mail servers before it arrives at its destination mail server. SMTP is the protocol that carries this message, and it works on top of TCP because it is a reliable protocol and provides sequencing and acknowledgments to ensure the e-mail message arrived successfully at its destination.

The user's e-mail client must be SMTP-compliant to be properly configured to use this protocol. The e-mail client provides an interface to the user so the user can create and modify messages as needed, and then the client passes the message off to the SMTP application layer protocol. So, to use the analogy of sending a letter via the post office, the e-mail client is the typewriter that a person uses to write the message, SMTP is the mail courier who picks up the mail and delivers it to the post office, and the post office is the mail server. The mail server has the responsibility of understanding where the message is heading and properly routing the message to that destination.

It is worth noting that basic SMTP doesn't include any security controls. This is why the IETF published Extended SMTP (ESMTP), which, among other features, allows servers to negotiate a TLS session in which to exchange the messages. This implementation, referred to as SMTP Secure (SMTPS), can provide authentication, confidentiality, and integrity protections for mail transfers.

The mail server is often referred to as an SMTP server. The most common SMTP server software in the world is Exim, which is an open-source mail transfer agent (MTA). SMTP works closely with two mail server protocols, POP and IMAP, which are explained in the following sections.

E-mail Threats

E-mail spoofing is a technique used by malicious users to forge an e-mail to make it appear to be from a legitimate source. Usually, such e-mails appear to be from known and trusted e-mail addresses when they are actually generated from a malicious source. This technique is widely used by attackers these days for spamming and phishing purposes. An attacker tries to acquire the target's sensitive information, such as username and password or bank account credentials. Sometimes, the e-mail messages contain a link of a known website when it is actually a fake website used to trick the user into revealing his information.

E-mail spoofing is done by modifying the fields of e-mail headers, such as the From, Return-Path, and Reply-To fields, so the e-mail appears to be from a trusted source. This results in an e-mail looking as though it is from a known e-mail address. Mostly the From field is spoofed, but some scams have modified the Reply-To field to the attacker's e-mail address. E-mail spoofing is caused by the lack of security features in SMTP. When SMTP technologies were developed, the concept of e-mail spoofing didn't exist, so countermeasures for this type of threat were not embedded into the protocol. A user could use an SMTP server to send e-mail to anyone from any e-mail address. We'll circle back to these threats when we describe e-mail security later in this section.

POP

Post Office Protocol (POP) is an Internet mail server protocol that supports incoming and outgoing messages. The current version is 3, so you'll also see it referred to as POP3. A mail server that uses POP, apart from storing and forwarding e-mail messages, works with SMTP to move messages between mail servers. By default, POP servers listen on TCP port 110.

A smaller organization may have only one POP server that holds all employee mailboxes, whereas larger organizations could have several POP servers, one for each department within the organization. There are also Internet POP servers that enable people all over the world to exchange messages. This system is useful because the messages are held on the mail server until users are ready to download their messages, instead of trying to push messages right to a person's computer, which may be down or offline.

The e-mail server can implement different authentication schemes to ensure an individual is authorized to access a particular mailbox, but this is usually handled through usernames and passwords. Connections to these clients can be encrypted using TLS by using the secure version of POP, known as POP3S, which typically listens on port 995.

IMAP

Internet Message Access Protocol (IMAP) is also an Internet protocol that enables users to access mail on a mail server (the default TCP port is 143). IMAP provides all the functionalities of POP, but has more capabilities. If a user is using POP, when he accesses his mail server to see if he has received any new messages, all messages are automatically downloaded to his computer. Once the messages are downloaded from the POP server, they are usually deleted from that server, depending upon the configuration. POP can cause frustration for mobile users because the messages are automatically pushed down to their computer or device and they may not have the necessary space to hold all the messages. This is especially true for mobile devices that can be used to access e-mail servers. This is also inconvenient for people checking their mail on other people's computers. If Christina checks her e-mail on Jessica's computer, all of Christina's new mail could be downloaded to Jessica's computer.

If a user uses IMAP instead of POP, she can download all the messages or leave them on the mail server within her remote message folder, referred to as a mailbox. The user can also manipulate the messages within this mailbox on the mail server as if the messages resided on her local computer. She can create or delete messages, search for specific messages, and set and clear flags. This gives the user much more freedom and keeps the messages in a central repository until the user specifically chooses to download all messages from the mail server.

IMAP is a store-and-forward mail server protocol that is considered POP's successor. IMAP also gives administrators more capabilities when it comes to administering and maintaining the users' messages. Just like SMTP and POP, IMAP can run over TLS, in which case the server listens for connections on TCP port 993.

E-mail Authorization

POP has the capability to integrate *Simple Authentication and Security Layer (SASL)*, a protocol-independent framework for performing authentication. This means that any protocol that knows how to interact with SASL can use its various authentication mechanisms without having to actually embed the authentication mechanisms within its code.

To use SASL, a protocol includes a command for identifying and authenticating a user to an authentication server and for optionally negotiating protection of subsequent protocol interactions. If its use is negotiated, a security layer is inserted between the protocol and the connection. The data security layer can provide data integrity, data confidentiality, and other services. SASL's design is intended to allow new protocols to reuse existing mechanisms without requiring redesign of the mechanisms and allows existing protocols to make use of new mechanisms without redesign of protocols.

The use of SASL is not unique just to POP; other protocols, such as IMAP, Internet Relay Chat (IRC), Lightweight Directory Access Protocol (LDAP), and SMTP, can also use SASL and its functionality.

Sender Policy Framework

A common way to deal with the problem of forged e-mail messages is by using *Sender Policy Framework (SPF)*, which is an e-mail validation system designed to prevent e-mail spam by detecting e-mail spoofing by verifying the sender's IP address. SPF allows administrators to specify which hosts are allowed to send e-mail from a given domain by creating a specific SPF record in DNS. Mail exchanges use DNS to check that mail from a given domain is being sent by a host sanctioned by that domain's administrators.

DomainKeys Identified Mail

We can also leverage public key infrastructure (PKI) to validate the origin and integrity of each message. The *DomainKeys Identified Mail (DKIM)* standard, codified in RFC 6376, allows e-mail servers to digitally sign messages to provide a measure of confidence for the receiving server that the message is from the domain it claims to be from. These digital signatures are normally invisible to the user and are just used by the servers sending and receiving the messages. When a DKIM-signed message is received, the server requests the sending domain's certificate through DNS and verifies the signature. As long as the private key is not compromised, the receiving server is assured that the message came from the domain it claims and that it has not been altered in transit.

Domain-Based Message Authentication

SPF and DKIM were brought together to define the Domain-based Message Authentication, Reporting and Conformance (DMARC) system. DMARC, which today is estimated to protect 80 percent of mailboxes worldwide, defines how domains communicate to the rest of the world whether they are using SPF or DKIM (or both). It also codifies the mechanisms by which receiving servers provide feedback to the senders on the results of their validation of individual messages. Despite significant advances in securing e-mail, phishing e-mail remains one of the most common and effective attack vectors.

Secure/Multipurpose Internet Mail Extensions

Multipurpose Internet Mail Extensions (MIME) is a technical specification indicating how multimedia data and e-mail binary attachments are to be transferred. The Internet has mail standards that dictate how mail is to be formatted, encapsulated, transmitted, and opened. If a message or document contains a binary attachment, MIME dictates how that portion of the message should be handled.

When an attachment contains an audio clip, graphic, or some other type of multimedia component, the e-mail client sends the file with a header that describes the file type. For example, the header might indicate that the MIME type is Image and that the subtype is jpeg. Although this information is in the header, many times, systems also use the file's extension to identify the MIME type. So, in the preceding example, the file's name might be stuff.jpeg. The user's system sees the extension .jpeg, or sees the data in the header field, and looks in its association list to see what program it needs to initialize to open this particular file. If the system has JPEG files associated with the Explorer application, then Explorer opens and presents the image to the user.

Sometimes systems either do not have an association for a specific file type or do not have the helper program necessary to review and use the contents of the file. When a file has an unassociated icon assigned to it, it might require the user to choose the Open With command and choose an application in the list to associate this file with that program. So when the user double-clicks that file, the associated program initializes and presents the file. If the system does not have the necessary program, the website might offer the necessary helper program, like Acrobat or an audio program that plays WAV files.

MIME is a specification that dictates how certain file types should be transmitted and handled. This specification has several types and subtypes, enables different computers to exchange data in varying formats, and provides a standardized way of presenting the data. So if Sean views a funny picture that is in GIF format, he can be sure that when he sends it to Debbie, it will look exactly the same.

Secure MIME (S/MIME) is a standard for encrypting and digitally signing e-mail and for providing secure data transmissions. S/MIME extends the MIME standard by providing support for the encryption of e-mail and attachments. The encryption and hashing algorithms can be specified by the user of the mail application, instead of having it dictated to them. S/MIME follows the Public Key Cryptography Standards (PKCS). It provides confidentiality through encryption algorithms, integrity through hashing algorithms, authentication through the use of X.509 public key certificates, and nonrepudiation through cryptographically signed message digests.

Multilayer Protocols

Not all protocols fit neatly within the layers of the OSI model. This is particularly evident among devices and networks that were never intended to interoperate with the Internet. For this same reason, they tend to lack robust security features aimed at protecting the availability, integrity, and confidentiality of the data they communicate. The problem is that as the Internet of old becomes the Internet of Things (IoT), these previously isolated devices and networks find themselves increasingly connected to a host of threats they were never meant to face.

As security professionals, we need to be aware of these nontraditional protocols and their implications for the security of the networks to which they are connected. In particular, we should be vigilant when it comes to identifying nonobvious cyber-physical systems. In December 2015, attackers were able to cut power to over 80,000 homes in Ukraine apparently by compromising the utilities' supervisory control and data acquisition (SCADA) systems in what is considered the first known blackout caused by a cyberattack. A few years later, in 2017, attackers were able to exploit a previously unknown vulnerability and reprogram a Schneider Electric safety instrumented system (SIS) at an undisclosed target, causing the facility to shut down. At the heart of most SCADA systems used by power and water utilities is a multilayer protocol known as DNP3.

Distributed Network Protocol 3

The *Distributed Network Protocol 3 (DNP3)* is a communications protocol designed for use in SCADA systems, particularly those within the power sector. It is not a general-purpose protocol like IP, nor does it incorporate routing functionality. SCADA systems typically have a very flat hierarchical architecture in which sensors and actuators are connected to remote terminal units (RTUs). The RTUs aggregate data from one or more of these devices and relay it to the SCADA master, which includes a human–machine interface (HMI) component. Control instructions and configuration changes are sent from the SCADA master to the RTUs and then on to the sensors and actuators.

At the time DNP3 was designed, there wasn't a need to route traffic among the components (most of which were connected with point-to-point circuits), so networking was not needed or supported in DNP3. Instead of using the OSI seven-layer model, its developers opted for a simpler three-layer model called the Enhanced Performance Architecture (EPA) that roughly corresponds to layers 2, 4, and 7 of the OSI model. There was no encryption or authentication, since the developers did not think network attacks were feasible on a system consisting of devices connected to each other and to nothing else.

Over time, SCADA systems were connected to other networks and then to the Internet for a variety of very valid business reasons. Unfortunately, security wasn't considered until much later. Encryption and authentication features were added as an afterthought, though not all implementations have been thus updated. Network segmentation is not always present either, even in some critical installations. Perhaps most concerning is the shortage of effective IPSs and IDSs that understand the interconnections between DNP3 and IP networks and can identify DNP3-based attacks.

Controller Area Network Bus

Another multilayer protocol that had almost no security features until very recently is the one that runs most automobiles worldwide. The *Controller Area Network (CAN) bus* is a protocol designed to allow microcontrollers and other embedded devices to communicate with each other on a shared bus. Over time, these devices have diversified so that today they can control almost every aspect of a vehicle's functions, including steering, braking, and throttling. CAN bus was never meant to communicate with anything outside the vehicle except for a mechanic's maintenance computer, so there never appeared to be a need for security features.

As automobiles started getting connected via Wi-Fi and cellular data networks, their designers didn't fully consider the new attack vectors this would introduce to an otherwise undefended system. That is, until Charlie Miller and Chris Valasek famously hacked a Jeep in 2015 by connecting to it over a cellular data network and bridging the head unit (which controls the sound system and GPS) to the CAN bus (which controls all the vehicle sensors and actuators) and causing it to run off a road. As automobiles become more autonomous, security of the CAN bus becomes increasingly important.

Modbus

Like CAN bus, the Modbus system was developed to prioritize functionality over security. A communications system created in the late 1970s by Modicon, now Schneider Electric, Modbus enables communications among SCADA devices quickly and easily. Since its inception, Modbus has quickly become the de facto standard for communications between programmable logic controllers (PLCs). But as security was not built in, Modbus offers little protection against attacks. An attacker residing on the network can simply collect traffic using a tool like Wireshark, find a target device, and issue commands directly to the device.

Converged Protocols

Converged protocols are those that started off independent and distinct from one another but over time converged to become one. How is this possible? Think about the phone and data networks. Once upon a time, these were two different entities and each had its own protocols and transmission media. For a while, in the 1990s, data networks sometimes rode over voice networks using data modems. This was less than ideal, which is why we flipped it around and started using data networks as the carrier for voice communications. Over time, the voice protocols converged onto the data protocols, which paved the way for Voice over IP (VoIP).

IP convergence, which addresses a specific type of converged protocols, is the transition of services from disparate transport media and protocols to IP. It is not hard to see that IP has emerged as the dominant standard for networking, so it makes sense that any new protocols would leverage this existing infrastructure rather than create a separate one.

Technically, the term *converged* implies that the two protocols became one. Oftentimes, however, the term is used to describe cases in which one protocol was originally independent of another but over time started being encapsulated (or tunneled) within that other one.

Encapsulation

We already saw (in Chapter 9) how encapsulation enables the transmission of data down the seven layers of the OSI reference model. We came across encapsulation again earlier in this chapter when we discussed techniques to tunnel (or encapsulate) one protocol's traffic inside some other protocol. The next two sections describe two more examples. It should be obvious that encapsulation can be helpful in architecting our networks, but it can also have significant security implications.

When we covered DNS tunneling, we saw another, less helpful application of encapsulation. Threat actors develop their own protocols for controlling compromised hosts and they can encapsulate those protocols within legitimate systems. It is important, therefore, to not assume that just because we have a network link that should be transporting data of a certain protocol, it won't have something else embedded in it. Whether encapsulation is malicious or benign, the point is that we need to be aware of what traffic should be where and have the means to inspect it to ensure we are not surprised.

Fiber Channel over Ethernet

Fibre Channel (FC) (also called Fiber Channel in the United States) was developed by the American National Standards Institute (ANSI) in 1988 as a way to connect supercomputers using optical fibers. FC is now used to connect servers to data storage devices in data centers and other high-performance environments. One of its best features is that it can support speeds of up to 128 Gbps over distances of up to 500 meters. (Distances of up to 50 km are possible at lower data rates.) While the speed and other features of FC are pretty awesome for data centers and storage area network (SAN) applications, the need to maintain both Ethernet and fiber-optic cabling adds costs and complexity to its use in enterprise environments.

Fibre Channel over Ethernet (FCoE) is a protocol encapsulation that allows FC frames to ride over Ethernet networks. Its use allows data centers to be almost exclusively wired using Ethernet cabling. It is important to note, however, that FCoE rides on top of Ethernet and is, therefore, a non-routable protocol. It is only intended for LAN environments where devices are in close proximity to each other and efficiency is essential.

Internet Small Computer Systems Interface

A much different approach to encapsulation is exemplified by the Internet Small Computer Systems Interface (iSCSI), which encapsulates SCSI data in TCP segments. SCSI is a set of technologies that allows peripherals to be connected to computers. The problem with the original SCSI is that it has limited range, which means that connecting a remote peripheral (e.g., camera or storage device) is not normally possible. The solution was to let SCSI ride on TCP segments so that a peripheral device could be literally anywhere in the world and still appear as local to a computer.

Network Segmentation

Once upon a time, networks were flat (i.e., almost everyone within an organization was in the same layer 2 broadcast domain) so that everyone could easily communicate with everyone else inside the "trusted" perimeter. Network defenses were mostly (sometimes solely) outward-facing. This led to the networks that were "crunchy on the outside but soft and chewy on the inside." Believe it or not, this was the design mantra for many organizations for many years. Eventually, they realized that this design was a really bad idea. For starters, they recognized that at least some attackers will get through their perimeter defenses. Also, they learned that insider threats could be just as dangerous as external ones, and these insiders would have no problem moving through the soft and chewy interior network. Furthermore, they realized that most networks no longer have a neat concept of "inside" and "outside." Instead, organizations increasingly rely on external systems such as those provided by cloud service providers.

Network segmentation is the practice of dividing networks into smaller subnetworks. An example is to divide the network by department, so that the finance department and marketing department are each in their own LAN. If they need to communicate directly, they have to go through a gateway (e.g., a router or firewall) that allows network administrators to block or detect suspicious traffic. This is a classic implementation of the zero trust security design principle.

The decision to segment a network begs a couple of questions. How many subnetworks should we have? Are more subnets better? There really is no one-size-fits-all answer, but generally, the smaller the subnetworks (and the more you have), the better. In fact, many organizations are implementing *micro-segmentation*, which is the practice of isolating individual assets (e.g., data servers) in their own protected network environment. Think of it as a subnet where the only devices are the protected asset and a security gateway.

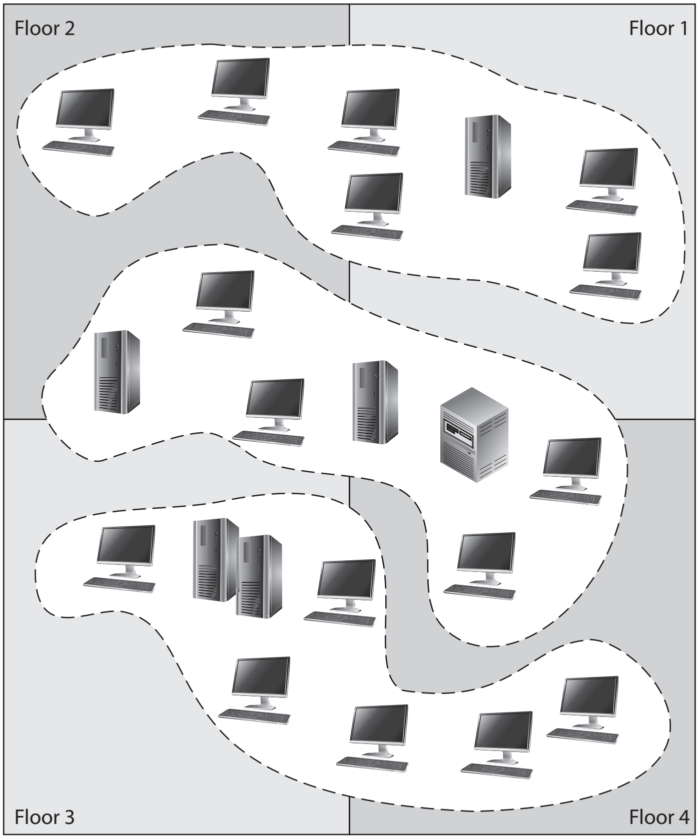
So, how do we go about segmenting networks? We can do it either physically (using devices like switches and routers) or logically (using virtualization software). We'll cover devices in detail in the next chapter, so let's turn our attention to the most important technologies that enable segmentation and micro-segmentation.

VLANs

One of the most commonly used technologies used to segment LANs is the *virtual local area network (VLAN)*. A LAN can be defined as a set of devices on the same layer 2 (data link layer) broadcast domain. This typically means hosts that are *physically* connected to the same layer 2 switches. A VLAN is a set of devices that *behave* as though they were all directly connected to the same switch, when in fact they aren't. This allows you to, for instance, ensure that all members of the finance team are on the same (virtual) LAN, even though they are scattered across multiple countries. The ability to segment networks of users in this manner is critical for both functional and security reasons.

Virtually all modern enterprise-grade switches have the capability to use VLANs. VLANs enable administrators to separate and group computers logically based on resource requirements, security, or business needs instead of the standard physical location of the systems. When repeaters, bridges, and routers are used, systems and resources are grouped in a manner dictated by their physical location. Figure 13-10 shows how computers that are physically located next to each other can be grouped logically into different VLANs. Administrators can form these groups based on the users' and organization's needs instead of the physical location of systems and resources.

Figure 13-10 VLANs enable administrators to manage logical networks.

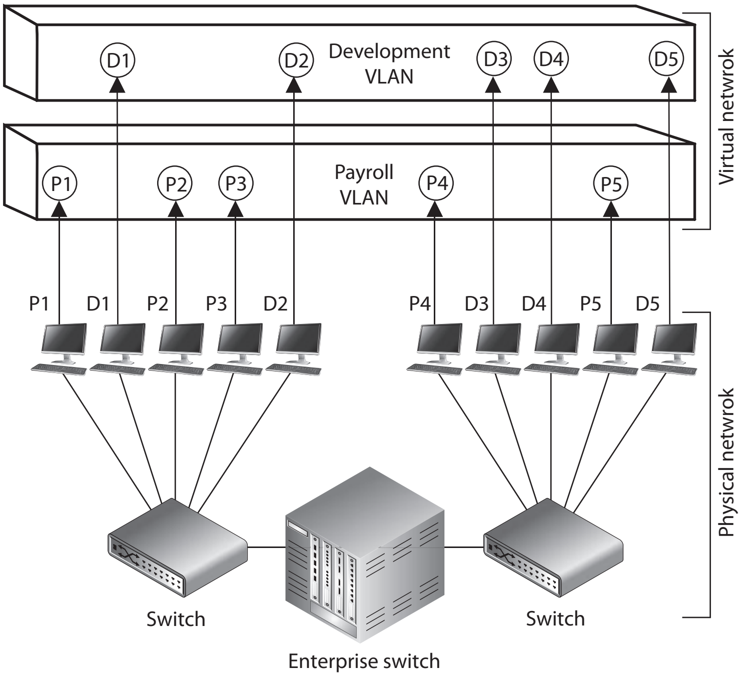


An administrator may want to segment the computers of all users in the marketing department in the same VLAN network, for example, so all users receive the same broadcast messages and can access the same types of resources. This arrangement could get tricky if a few of the users are located in another building or on another floor, but VLANs provide the administrator with this type of flexibility. VLANs also enable an administrator to apply particular security policies to respective zones or segments. This way, if tighter security is required for the payroll department, for example, the administrator can develop a policy, add all payroll systems to a specific VLAN, and apply the security policy only to the payroll VLAN.

A VLAN exists on top of the physical network, as shown in Figure 13-11. Each Ethernet frame is prepended with a VLAN identifier (VID), which is a 12-bit field. This means that we can define up to 4,095 VLANs in the same network. (The first and last VID values are reserved.) If workstation P1 wants to communicate with workstation D1, the message has to be routed—even though the workstations are physically next to each other—because they are on different logical networks.

NOTE The IEEE standard that defines how VLANs are to be constructed and how tagging should take place to allow for interoperability is IEEE 802.1Q.

Figure 13-11 VLANs exist on a higher level than the physical network and are not bound to it.



While VLANs are used to segment traffic, attackers can still gain access to traffic that is supposed to be "walled off" in another VLAN segment. *VLAN hopping attacks* allow attackers to gain access to traffic in various VLAN segments. An attacker can have a system act as though it is a switch. The system understands the tagging values being used in the network and the trunking protocols and can insert itself between other VLAN devices and gain access to the traffic going back and forth. This is called a *switch spoofing attack*. An attacker can also insert VLAN tags to manipulate the control of traffic at the data link layer in what is known as a *double tagging attack*. Proper configuration of all switches mitigates VLAN hopping attacks.

Virtual eXtensible Local Area Network

VLANs, however, have some significant limitations. For starters, remember that you're limited to around 4,000 VLANs because the VID is 12 bits. While this sounds like a lot, it really isn't if you happen to be a cloud-based service provider supporting hundreds of customers. Another challenge is that VLANs are layer 2 constructs separated by layer 3 routers. This means that all the hosts on a given VLAN must be on the same port of the same router. In other words, if the hosts are in different countries, it becomes really hard to join them to the same VLAN.

The *Virtual eXtensible Local Area Network (VxLAN)* is a network virtualization technology that encapsulates layer 2 frames onto UDP (layer 4) datagrams for distribution anywhere in the world. Whereas VLANs have VIDs, VxLANs have a virtual network identifier (VNI) that is 24 bits long, which gives us over 16 million segments. VxLANs are mostly used in cloud environments where hosts and networks are virtualized.

VxLANs are overlay networks on top of UDP/IP underlay networks. Each network switch or router that is part of a VxLAN has a *virtual tunnel end point (VTEP)* that provides the interface between the underlay and overlay networks. When a VTEP receives a frame, it establishes a virtual tunnel on the overlay network connecting it to the destination VTEP just long enough to deliver the frame. The VTEP encapsulates this overlay frame in UDP datagrams that are then passed to the underlay network for delivery.

Software-Defined Networks

Software-defined networking (SDN) is an approach to networking that relies on distributed software to provide unprecedented agility and efficiency. Using SDN, it becomes much easier to dynamically route traffic to and from newly provisioned services and platforms. This means a new server can be quickly provisioned using a cloud service provider in response to a spike in service requests and the underlying network can just as quickly adapt to the new traffic patterns. It also means that a service or platform can be quickly moved from one location to another and the SDN will just as quickly update traffic-flow rules in response to this change. Unsurprisingly, the three biggest drivers to the adoption of SDN are the growth in cloud computing, big data, and mobile computing.

How does SDN differ from traditional networking? Whereas traditional networking relies on network devices that coordinate with one another in a mostly decentralized manner, SDN centralizes the configuration and control of devices. In a decentralized environment, it takes time for routers to converge onto (or agree on) good routes. These devices must normally be manually configured whenever any changes take place, which is also a time-consuming task. In SDN, on the other hand, all changes are pushed out to the devices either reactively (i.e., in response to requests from the devices) or proactively (i.e., because the admins know a change is being made, such as the addition of 100 servers). Because it is centrally controlled, the SDN approach allows traffic to be routed much more efficiently and securely. Perhaps the most important element of SDN is the abstraction of control and forwarding planes.

Control and Forwarding Planes

The *control plane* is where the internetwork routing decisions are being made. Think of this as the part of your router that runs the routing protocol, such as Open Shortest Path First (OSPF). (The analogy is not perfect, but it is useful for now.) The control plane is responsible for discovering the topology of neighboring networks and maintaining a table of routes for outbound packets. Since most networks are pretty dynamic places in which congestion along different routes is always changing, the control plane is a pretty dynamic place as well. New routes are routinely being discovered, just as old routes are dropped or at least flagged as slow or expensive. As you can see, the control plane is mostly interested in effects that are more than one hop away.

The *forwarding plane*, by contrast, is where traffic forwarding decisions are made. Think of this as that part of your router that decides (very quickly) that a packet received on network interface eth0 needs to be forwarded to network interface eth3. How does the forwarding plane decide this? By using the products developed by the control plane. The control plane is the strategic, methodical planner of traffic routing, while the forwarding plane is the tactical, fast executioner of those plans. Unsurprisingly, the forwarding plane is typically implemented in hardware such as an application-specific integrated chip (ASIC).

NOTE Because traditional routing decisions are made by the controller in an SDN architecture, the network devices behave (and are referred to) as switches.

In a traditional network architecture, each networking device has its own control plane and its own forwarding plane, both of which run on some sort of proprietary operating system (e.g., Cisco IOS). The normal way of reconfiguring these traditional devices is via a terminal connection of some sort. This means that an administrator must remotely log into each device in order to change its configuration. Let's suppose that we want to support a distinct QoS for a new user. In order to do this, we'd modify the configuration in each networking device that would be involved in providing services to this user. Even assuming that we are able to do this without making any mistakes, we still face the onerous task of manually changing these parameters whenever the terms of the contract change, or when equipment is replaced or upgraded, or when the network architecture changes. There are exceptions to these challenges, of course, but the point is that making frequent, granular configuration changes is tough.

What About Automation?

One of the challenges of network administration is that most network devices (apart from those that support SDN) do not have comprehensive mechanisms for programmatically and remotely changing the configuration of the device. This is why administrators have to manually log into each device and update the configuration. Reading information is easier because these devices typically support SNMP, but writing meaningful changes to the devices almost always requires manual interaction or some third-party tool that comes with its own set of constraints.

Further complicating the issue of making dynamic changes, vendors typically use their own proprietary operating system, which makes it harder to write a script that makes the same changes to all devices in heterogeneous environments that implement products from multiple vendors. This is the reason why many organizations implement homogeneous network architectures in which all the devices are manufactured by the same vendor. A big downside of this homogeneity is that it leads to vendor lockdown because it is hard (and expensive) to change vendors when that means you must change every single device on your network. Furthermore, homogeneity is bad for security, because an exploit that leverages a vulnerability in a network operating system will likely affect every device in a homogeneous network.

In SDN, by contrast, the control plane is implemented in a central node that is responsible for managing all the devices in the network. For redundancy and efficiency, this node can actually be a federation of nodes that coordinate their activities with one another. The network devices are then left to do what they do best: forward packets very efficiently. So the forwarding plane lives in the network devices and the control plane lives in a centralized SDN controller. This allows us to abstract the network devices (heterogeneous or otherwise) from the applications that rely on them to communicate in much the same way Windows abstracts the hardware details from the applications running on a workstation.

Approaches to SDN

The concept of network abstraction is central to all implementations of SDN. The manner in which this abstraction is implemented, however, varies significantly among flavors of SDN. There are at least three common approaches to SDN, each championed by a different community and delivered primarily through a specific technology:

- **Open** The SDN approach championed by the Open Networking Foundation (ONF) (https://opennetworking.org) is, by most accounts, the most common. It relies on open-source code and standards to develop the building blocks of an SDN solution. The controller communicates with the switches using OpenFlow, a standardized, open-source communications interface between controllers and network devices in an SDN architecture. OpenFlow allows the devices implementing the forwarding plane to provide information (such as utilization data) to the controller, while allowing the controller to update the flow tables (akin to traditional routing tables) on the devices. Applications communicate with the controller using the RESTful or Java APIs.

- **API** Another approach to SDN, and one that is championed by Cisco, is built on the premise that OpenFlow is not sufficient to fully leverage the promise of SDN in the enterprise. In addition to OpenFlow, this approach leverages a rich API on proprietary switches that allows greater control over traffic in an SDN. Among the perceived shortcomings that are corrected are the inability of OpenFlow to do deep packet inspection and manipulation and its reliance on a centralized control plane. This proprietary API approach to SDN is seen as enriching rather than replacing ONF's SDN approach.
- **Overlays** Finally, one can imagine a virtualized network architecture as an overlay on a traditional one. In this approach, we virtualize all network nodes, including switches, routers, and servers, and treat them independently of the physical networks upon which this virtualized infrastructure exists. The SDN exists simply as a virtual overlay on top of a physical (underlay) network.

Software-Defined Wide Area Network

Software-defined wide area networking (SD-WAN) is the use of software (instead of hardware) to control the connectivity, management, and services between distant sites. Think of it as SDN applied to WANs instead of LANs. Similarly to SDN, SD-WAN separates the control plane from the forwarding plane. This means that network links, whether they are leased lines or 5G wireless, are better utilized. Also, since the control plane is centralized, security policies can be consistently applied throughout.

Another advantage of SD-WANs is that they are application-aware, meaning they know the difference between supporting video conferencing (low latency, loss tolerance), supporting file transfers (latency tolerance, loss intolerant), or supporting any other sort of traffic. This means SD-WANs use the right path for the traffic and are able to switch things around as links become congested or degraded.

Chapter Review

Securing our networks is a lot more effective if we first understand the underlying technologies and then apply secure design principles to their selection and integration. This chapter built on the foundations of the previous two chapters to show common approaches to building and operating secure networking architectures. We focused our attention on network encryption and service security techniques but also covered how to deal with dispersed networks and those with cloud service components. A key aspect of our discussion was the application of the secure design principles at multiple points. We'll continue this theme in the next chapter as we talk about securing the components of our networks.

Quick Review

- Link encryption encrypts all the data along a specific communication path.
- End-to-end encryption (E2EE) occurs at the session layer (or higher) and does not encrypt routing information, enabling attackers to learn more about a captured packet and where it is headed.
- Transport Layer Security (TLS) is an E2EE protocol that provides confidentiality and data integrity for network communications.
- Secure Sockets Layer (SSL) is the predecessor of TLS and is deprecated and considered insecure.
- A virtual private network (VPN) is a secure, private connection through an untrusted network.
- The Point-to-Point Tunneling Protocol (PPTP) is an obsolete and insecure means of providing VPNs.
- The Layer 2 Tunneling Protocol (L2TP) tunnels PPP traffic over various network types (IP, ATM, X.25, etc.) but does not encrypt the user traffic.
- Internet Protocol Security (IPSec) is a suite of protocols that provides authentication, integrity, and confidentiality protections to data at the network layer.
- TLS can be used to provide VPN connectivity at layer 5 in the OSI model.
- A web service is client/server system in which clients and servers communicate using HTTP over a network such as the Internet.
- A service-oriented architecture (SOA) describes a system as a set of interconnected but self-contained components that communicate with each other and with their clients through standardized protocols.
- Application programming interfaces (APIs) establish a "language" that enables a system component to make a request from another component and then interpret that second component's response.
- The Hypertext Transfer Protocol (HTTP) is a TCP/IP-based communications protocol used for transferring data between a server and a client in a connectionless and stateless manner.
- A uniform resource identifier (URI) uniquely identifies a resource on the Internet.
- HTTP Secure (HTTPS) is HTTP running over TLS.
- The Simple Object Access Protocol (SOAP) is a messaging protocol that uses XML over HTTP to enable clients to invoke processes on a remote host in a platform-agnostic way.
- SOAP security is enabled by a set of protocol extensions called the Web Services Security (WS-Security or WSS) specification, which provides message confidentiality, integrity, and authentication.

- Representational State Transfer (REST) is an architectural pattern used to develop web services without using SOAP.
- A domain generation algorithm (DGA) produces seemingly random domain names in a way that is predictable by anyone who knows the algorithm.
- DNS tunneling is the practice of encoding messages in one or a series of DNS queries or responses for exfiltrating or infiltrating data into an environment.
- DNS reflection attacks involve sending a query to a server while spoofing the source address to be that of the intended target.
- A DNS amplification attack is characterized by small queries that result in very much larger responses.
- Domain Name System Security Extensions (DNSSEC) is a set of IETF standards that ensures the integrity of DNS records but not their confidentiality or availability.
- DNS over HTTPS (DoH) is a (yet to be ratified) approach to protecting the privacy and confidentiality of DNS queries by sending them over HTTPS/TCP /IP instead of unsecured UDP/IP.
- E-mail spoofing is a technique used by malicious users to forge an e-mail to make it appear to be from a legitimate source.
- Simple Authentication and Security Layer (SASL) is a protocol-independent framework for performing authentication that is typically used in POP3 e-mail systems.
- The Sender Policy Framework (SPF) is an e-mail validation system designed to prevent e-mail spam by detecting e-mail spoofing by verifying the sender's IP address.
- The DomainKeys Identified Mail (DKIM) standard allows e-mail servers to digitally sign messages to provide a measure of confidence for the receiving server that the message is from the domain it claims to be from.
- Domain-based Message Authentication, Reporting and Conformance (DMARC) systems incorporate both SPF and DKIM to protect e-mail.
- Secure MIME (S/MIME) is a standard for encrypting and digitally signing e-mail and for providing secure data transmissions.
- The Distributed Network Protocol 3 (DNP3) is a multilayer communications protocol designed for use in SCADA systems, particularly those within the power sector.
- The Controller Area Network (CAN) bus is a multilayer protocol designed to allow microcontrollers and other embedded devices to communicate with each other on a shared bus.
- Converged protocols are those that started off independent and distinct from one another but over time converged to become one.

- Fibre Channel over Ethernet (FCoE) is a protocol encapsulation that allows Fibre Channel (FC) frames to ride over Ethernet networks.
- The Internet Small Computer Systems Interface (iSCSI) protocol encapsulates SCSI data in TCP segments so that computer peripherals could be located at any physical distance from the computer they support.
- Network segmentation is the practice of dividing networks into smaller subnetworks.
- A virtual LAN (VLAN) is a set of devices that behave as though they were all directly connected to the same switch, when in fact they aren't.
- Virtual eXtensible LAN (VxLAN) is a network virtualization technology that encapsulates layer 2 frames onto UDP (layer 4) datagrams for distribution anywhere in the world.
- Software-defined networking (SDN) is an approach to networking that relies on distributed software to separate the control and forwarding planes of a network.
- Software-defined wide area networking (SD-WAN) is the use of software (instead of hardware) to control the connectivity, management, and services between distant sites in a manner that is similar to SDN but applied to WANs.

Questions

Please remember that these questions are formatted and asked in a certain way for a reason. Keep in mind that the CISSP exam is asking questions at a conceptual level. Questions may not always have the perfect answer, and the candidate is advised against always looking for the perfect answer. Instead, the candidate should look for the best answer in the list.

1. Which of the following provides secure end-to-end encryption?
   - **A.** Transport Layer Security (TLS)
   - **B.** Secure Sockets Layer (SSL)
   - **C.** Layer 2 Tunneling Protocol (L2TP)
   - **D.** Domain Name System Security Extensions (DNSSEC)
2. Which of the following can take place if an attacker is able to insert tagging values into network- and switch-based protocols with the goal of manipulating traffic at the data link layer?
   - **A.** Open relay manipulation
   - **B.** VLAN hopping attack
   - **C.** Hypervisor denial-of-service attack
   - **D.** DNS tunneling

3. Which of the following provides an incorrect definition of the specific component or protocol that makes up IPSec?
   - A. Authentication Header protocol provides data integrity, data origin authentication, and protection from replay attacks.
   - B. Encapsulating Security Payload protocol provides confidentiality, data origin authentication, and data integrity.
   - C. Internet Security Association and Key Management Protocol provides a framework for security association creation and key exchange.
   - D. Internet Key Exchange provides authenticated keying material for use with encryption algorithms.
4. Alice wants to send a message to Bob, who is several network hops away from her. What is the best approach to protecting the confidentiality of the message?
   - A. PPTP
   - B. S/MIME
   - C. Link encryption
   - D. SSH
5. Which technology would best provide confidentiality to a RESTful web service?
   - A. Web Services Security (WS-Security)
   - B. Transport Layer Security (TLS)
   - C. HTTP Secure (HTTPS)
   - D. Simple Object Access Protocol (SOAP)
6. Which of the following protections are provided by Domain Name System Security Extensions (DNSSEC)?
   - A. Confidentiality and integrity
   - B. Integrity and availability
   - C. Integrity and authentication
   - D. Confidentiality and authentication
7. Which approach provides the best protection against e-mail spoofing?
   - A. Internet Message Access Protocol (IMAP)
   - B. Domain-based Message Authentication, Reporting and Conformance (DMARC)
   - C. Sender Policy Framework (SPF)
   - D. DomainKeys Identified Mail (DKIM)

8. Which of the following is a multilayer protocol developed for use in supervisory control and data acquisition (SCADA) systems?

   A. Controller Area Network (CAN) bus

   B. Simple Authentication and Security Layer (SASL)

   C. Control Plane Protocol (CPP)

   D. Distributed Network Protocol 3 (DNP3)

9. All of the following statements are true of converged protocols *except* which one?

   A. Distributed Network Protocol 3 (DNP3) is a converged protocol.

   B. Fibre Channel over Ethernet (FCoE) is a converged protocol.

   C. IP convergence addresses a specific type of converged protocols.

   D. The term includes certain protocols that are encapsulated within each other.

10. Suppose you work at a large cloud service provider that has thousands of customers around the world. What technology would best support segmentation of your customers' environments?

    A. Virtual local area network (VLAN)

    B. Virtual eXtensible Local Area Network (VxLAN)

    C. Software-defined wide area networking (SD-WAN)

    D. Layer 2 Tunneling Protocol (L2TP)

Answers

1. **A.** TLS and SSL are the only two answers that provide end-to-end encryption, but SSL is insecure, so it's not a good answer.

2. **B.** VLAN hopping attacks allow attackers to gain access to traffic in various VLAN segments. An attacker can have a system act as though it is a switch. The system understands the tagging values being used in the network and the trunking protocols and can insert itself between other VLAN devices and gain access to the traffic going back and forth. Attackers can also insert tagging values to manipulate the control of traffic at this data link layer.

3. **D.** Authentication Header protocol provides data integrity, data origin authentication, and protection from replay attacks. Encapsulating Security Payload protocol provides confidentiality, data origin authentication, and data integrity. Internet Security Association and Key Management Protocol provides a framework for security association creation and key exchange. Internet Key Exchange provides authenticated keying material for use with ISAKMP.

4. **B.** Secure Multipurpose Internet Mail Extensions (S/MIME) is a standard for encrypting and digitally signing e-mail and for providing secure data transmissions using public key infrastructure (PKI).

5. **C.** Either TLS or HTTPS would be a correct answer, but since web services in general and RESTful ones in particular require HTTP, HTTPS is the best choice. Keep in mind that you are likely to come across similar questions where multiple answers are correct but only one is best. SOAP is an alternative way to deliver web services and uses WS-Security for confidentiality.
6. **C.** Domain Name System Security Extensions (DNSSEC) is a set of IETF standards that ensures the integrity and authenticity of DNS records but not their confidentiality or availability.
7. **B.** Domain-based Message Authentication, Reporting and Conformance (DMARC) systems incorporate both SPF and DKIM to protect e-mail. IMAP does not have any built-in protections against e-mail spoofing.
8. **D.** DNP3 is a multilayer communications protocol designed for use in SCADA systems, particularly those within the power sector.
9. **A.** DNP3 is a multilayer communications protocol that was designed for use in SCADA systems and has not converged with other protocols. All other statements are descriptive of converged protocols.
10. **B.** Since there are thousands of customers to support, VxLAN is the best choice because it can support over 16 million subnetworks. Traditional VLANs are capped at just over 4,000 subnetworks, which would not be able to provide more than a few segments to each customer.

This page intentionally left blank

CHAPTER 14

Network Components

This chapter presents the following:

- Transmission media
- Network devices
- Endpoint security
- Content distribution networks

The hacker didn't succeed through sophistication. Rather he poked at obvious places, trying to enter through unlocked doors. Persistence, not wizardry, let him through.

—Clifford Stoll,
The Cuckoo's Egg

In the previous chapter, we covered how to defend our networks. Let's now talk about securing the components of those networks. We need to pay attention to everything from the cables, to the network devices, to the endpoints, because our adversaries will poke at all of it, looking for ways to get in. We (defenders) have to get it right all the time; they (attackers) only need to find that one chink in our armor to compromise our systems. In this chapter, we focus on physical devices. In the next chapter, we'll drill into the software systems that run on them.

Transmission Media

We've already talked a fair bit about the protocols that allow us to move data from point A to point B, but we haven't really covered what actually carries this information. A transmission medium is a physical thing through which data is moved. If we are speaking with each other, our vocal chords create vibrations in the air that we expel from our lungs, in which case the air is the transmission medium. Broadly speaking, we use three different types of transmission media:

- **Electrical wires** Encode information as changes in the voltage level of an electric current. Typically, we use cables, which are two or more wires encased within a sheath.
- **Optical fibers** Transmit data that is encoded in the wavelength (color), phase, or polarization of the light. The light is generated by either an LED or a laser diode. As with electrical wires, we usually bundle multiple fibers into cables for longer distances.

- **Free space** The medium we use for wireless communications, covered in Chapter 12. Any electromagnetic signal can travel through free space even outside our atmosphere. We tend to use mostly radio signals in free space, but every now and then you may encounter a system that uses light, such as infrared laser beams.

Types of Transmission

Physical data transmission can happen in different ways (analog or digital); can use different synchronization schemes (synchronous or asynchronous); can use either one sole channel over a transmission medium (baseband) or several different channels over a transmission medium (broadband); and can take place as electrical voltage, radio waves, or optical signals. These transmission types and their characteristics are described in the following sections.

Analog vs. Digital

A *signal* is just some way of moving information in a physical format from one point to another point. You can signal a message to another person through nodding your head, waving your hand, or giving a wink. Somehow you are transmitting data to that person through your signaling method. In the world of technology, we have specific carrier signals that are in place to move data from one system to another system. The carrier signal is like a horse, which takes a rider (data) from one place to another place. Data can be transmitted through analog or digital signaling formats. If you are moving data through an analog transmission technology (e.g., radio), then the data is represented by the characteristics of the waves that are carrying it. For example, a radio station uses a transmitter to put its data (music) onto a radio wave that travels all the way to your antenna. The information is stripped off by the receiver in your radio and presented to you in its original format—a song. The data is encoded onto the carrier signal and is represented by various amplitude and frequency values, as shown in Figure 14-1.

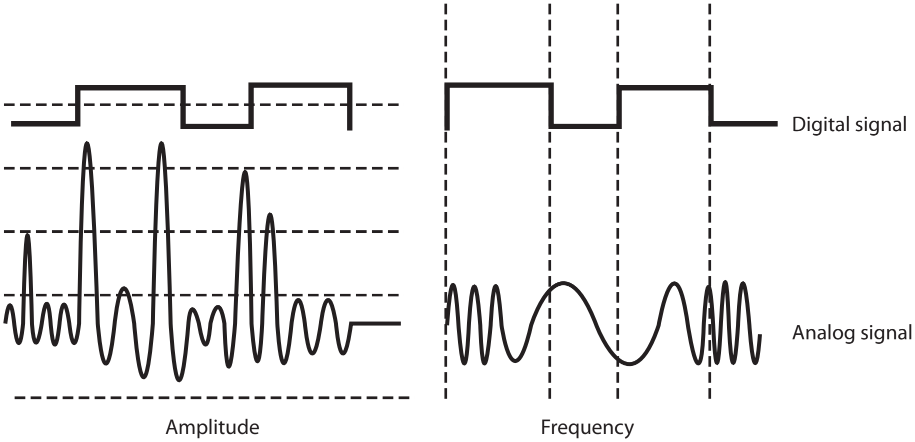


Figure 14-1 Analog signals are measured in amplitude and frequency, whereas digital signals represent binary digits as electrical pulses.

Data being represented in wave values (analog) is different from data being represented in discrete voltage values (digital). As an analogy, compare an analog clock and a digital clock. An analog clock has hands that continuously rotate on the face of the clock. To figure out what time it is, you have to interpret the position of the hands and map their positions to specific values. So you have to know that if the small hand is on the number 1 and the large hand is on the number 6, this actually means 1:30. The individual and specific location of the hands corresponds to a value. A digital clock does not take this much work. You just look at it and it gives you a time value in the format of hour:minutes. There is no mapping work involved with a digital clock because it provides you with data in clear-cut formats.

An analog clock can represent different values as the hands move forward—1:35 and 1 second, 1:35 and 2 seconds, 1:35 and 3 seconds. Each movement of the hands represents a specific value just like the individual data points on a wave in an analog transmission. A digital clock provides discrete values without having to map anything. The same is true with digital transmissions: the values are almost always binary, meaning they are either a 1 or a 0—no need for mapping to find the actual value.

Computers have always worked in a binary manner (1 or 0). When our telecommunication infrastructure was purely analog, each system that needed to communicate over a telecommunication line had to have a modem (modulator/demodulator), which would modulate the digital data into an analog signal. The sending system's modem would modulate the data on to the signal, and the receiving system's modem would demodulate the data off the signal.

Digital signals are more reliable than analog signals over a long distance and provide a clear-cut and efficient signaling method because the voltage is either on (1) or not on (0), compared to interpreting the waves of an analog signal. Extracting digital signals from a noisy carrier is relatively easy. It is difficult to extract analog signals from background noise because the amplitudes and frequencies of the waves slowly lose form. This is because an analog signal could have an infinite number of values or states, whereas a digital signal exists in discrete states. A digital signal is a square wave, which does not have all of the possible values of the different amplitudes and frequencies of an analog signal. Digital systems can implement compression mechanisms to increase data throughput, provide signal integrity through repeaters that "clean up" the transmissions, and multiplex different types of data (voice, data, video) onto the same transmission channel.

Asynchronous vs. Synchronous

Analog and digital transmission technologies deal with the characteristics of the physical carrier on which data is moved from one system to another. Asynchronous and synchronous transmission types are similar to the cadence rules we use for conversation *synchronization*. Asynchronous and synchronous network technologies provide synchronization rules to govern how systems communicate to each other. If you have ever spoken over a satellite phone, you have probably experienced problems with communication synchronization. Commonly, when two people are new to using satellite phones, they do not allow for the necessary delay that satellite communication requires, so they "speak over" one another. Once they figure out the delay in the connection, they resynchronize their timing so that only one person's data (voice) is transmitting at one time, enabling each person to properly understand the full conversation. Proper pauses frame your words in a way to make them understandable.

Synchronization through communication also happens when we write messages to each other. Properly placed commas, periods, and semicolons provide breaks in text so that the person reading the message can better understand the information. If you see "stickwithmekidandyouwillweardiamonds" without the proper punctuation, it is more difficult for you to understand. This is why we have grammar rules. If someone writes a letter to you that starts from the bottom and right side of a piece of paper, and that person does not inform you of this unconventional format, you will not be able to read the message properly, at least initially.

Technological communication protocols also have their own grammar and synchronization rules when it comes to the transmission of data. If two systems are communicating over a network protocol that employs asynchronous timing, they use start and stop bits. The sending system sends a "start" bit, then sends its character, and then sends a "stop" bit. This happens for the whole message. The receiving system knows when a character is starting and stopping; thus, it knows how to interpret each character of the message. This is akin to our previous example of using punctuation marks in written communications to convey pauses. If the systems are communicating over a network protocol that uses synchronous timing, then they don't add start and stop bits. The whole message is sent without artificial breaks, but with a common timing signal that allows the receiver to know how to interpret the information without these bits. This is similar to our satellite phone example in which we use a timing signal (i.e., we count off seconds in our head) to ensure we don't talk over the other person's speech.

If two systems are going to communicate using a synchronous transmission technology, they do not use start and stop bits, but the synchronization of the transfer of data takes place through a timing sequence, which is initiated by a clock pulse.

It is the data link protocol that has the synchronization rules embedded into it. So when a message goes down a system's network stack, if a data link protocol, such as High-level Data Link Control (HDLC), is being used, then a clocking sequence is in place. (The receiving system must also be using this protocol so that it can interpret the data.) If the message is going down a network stack and a protocol such as Asynchronous Transfer Mode (ATM) is at the data link layer, then the message is framed with start and stop indicators.

Data link protocols that employ synchronous timing mechanisms are commonly used in environments that have systems that transfer large amounts of data in a predictable manner (i.e., data center environment). Environments that contain systems that send data in a nonpredictable manner (i.e., Internet connections) commonly have systems with protocols that use asynchronous timing mechanisms.

So, synchronous communication protocols transfer data as a stream of bits instead of framing it in start and stop bits. The synchronization can happen between two systems using a clocking mechanism, or a signal can be encoded into the data stream to let the receiver synchronize with the sender of the message. This synchronization needs to take place before the first message is sent. The sending system can transmit a digital clock pulse to the receiving system, which translates into, "We will start here and work in this type of synchronization scheme." Many modern bulk communication systems,

Asynchronous	Synchronous
Simpler, less costly implementation	More complex, costly implementation
No timing component	Timing component for data transmission synchronization
Parity bits used for error control	Robust error checking, commonly through cyclic redundancy checking (CRC)
Used for irregular transmission patterns	Used for high-speed, high-volume transmissions
Each byte requires three bits of instruction (start, stop, parity)	Minimal protocol overhead compared to asynchronous communication

Table 14-1 Main Differences Between Asynchronous and Synchronous Transmissions

such as high-bandwidth satellite links, use Global Positioning System (GPS) clock signals to synchronize their communications without the need to include a separate channel for timing.

Table 14-1 provides an overview of the differences between asynchronous and synchronous transmissions.

Broadband vs. Baseband

As you read, analog transmission means that data is being moved as waves, and digital transmission means that data is being moved as discrete electric pulses. Synchronous transmission means that two devices control their conversations with a clocking mechanism, and asynchronous means that systems use start and stop bits for communication synchronization. Now let's look at how many individual communication sessions can take place at one time.

A *baseband* technology uses the entire communication channel for its transmission, whereas a *broadband* technology divides the communication channel into individual and independent subchannels so that different types of data can be transmitted simultaneously. Baseband permits only one signal to be transmitted at a time, whereas broadband carries several signals over different subchannels. For example, a coaxial cable TV (CATV) system is a broadband technology that delivers multiple television channels over the same cable. This system can also provide home users with Internet access, but this data is transmitted at a different frequency range than the TV channels.

As an analogy, baseband technology only provides a one-lane highway for data to get from one point to another point. A broadband technology provides a data highway made up of many different lanes, so that not only can more data be moved from one point to another point, but different types of data can travel over the individual lanes.

Any transmission technology that "chops up" one communication channel into multiple channels is considered broadband. The communication channel is usually a specific range of frequencies, and the broadband technology provides delineation between these frequencies and provides techniques on how to modulate the data onto the individual subchannels. To continue with our analogy, we could have one large highway that *could* fit eight individual lanes—but unless we have something that defines

How Do These Technologies Work Together?

If you are new to networking, it can be hard to understand how the OSI model, analog and digital, synchronous and asynchronous, and baseband and broadband technologies interrelate and differentiate. You can think of the OSI model as a structure to build different languages. If you and Luigi are going to speak to each other in English, you have to follow the rules of this language to be able to understand each other. If you are going to speak French, you still have to follow the rules of that language (OSI model), but the individual letters that make up the words are in a different order. The OSI model is a generic structure that can be used to define many different "languages" for devices to be able to talk to each other. Once you and Luigi agree that you are going to communicate using English, you can *speak* your message to Luigi, and thus your words move over continuous airwaves (analog). Or you can choose to send your message to Luigi through Morse code, which uses individual discrete values (digital). You can send Luigi all of your words with no pauses or punctuation (synchronous) or insert pauses and punctuation (asynchronous). If you are the only one speaking to Luigi at a time, this would be analogous to baseband. If ten people are speaking to Luigi at one time, this would be broadband.

these lanes and have rules for how these lanes are used, this is a baseband connection. If we take the same highway and lay down painted white lines, post traffic signs, add on and off ramps, and establish rules that drivers have to follow, now we are talking about broadband.

A digital subscriber line (DSL) uses one single phone line and constructs a set of high-frequency channels for Internet data transmissions. A cable modem uses the available frequency spectrum that is provided by a cable TV carrier to move Internet traffic to and from a household. Mobile broadband devices implement individual channels over a cellular connection, and Wi-Fi broadband technology moves data to and from an access point over a specified frequency set. The point is that there are different ways of cutting up one channel into subchannels for higher data transfer and that they provide the capability to move different types of traffic at the same time.

Cabling

The different types of transmission techniques we just covered eventually end up being used to send signals over either a cable or free space. We already covered wireless communications in Chapter 12, so let's talk about cabling now.

Electrical signals travel as currents through cables and can be negatively affected by many factors within the environment, such as motors, fluorescent lighting, magnetic forces, and other electrical devices. These items can corrupt the data as it travels through the cable, which is why cable standards are used to indicate cable type, shielding, transmission rates, and maximum distance a particular type of cable can be used.

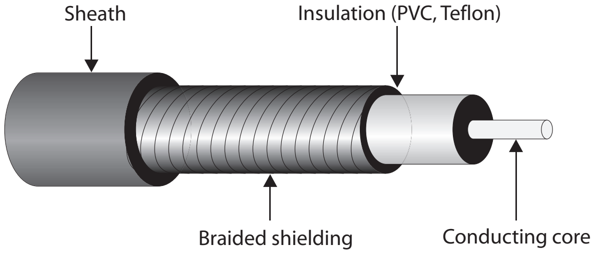


Figure 14-2
Coaxial cable

Coaxial Cable

Coaxial cable has a copper core that is surrounded by a shielding layer and grounding wire, as shown in Figure 14-2. This is all encased within a protective outer jacket. Compared to twisted-pair cable, coaxial cable is more resistant to electromagnetic interference (EMI), provides a higher bandwidth, and supports the use of longer cable lengths. So, why is twisted-pair cable more popular? Twisted-pair cable is cheaper and easier to work with, and the move to switched environments that provide hierarchical wiring schemes has overcome the cable-length issue of twisted-pair cable.

Coaxial cabling is used as a transmission line for radio frequency signals. If you have cable TV, you have coaxial cabling entering your house and the back of your TV. The various TV channels are carried over different radio frequencies. Modems allow us to use some of the "empty" TV frequencies for Internet connectivity.

Twisted-Pair Cable

Twisted-pair cabling has insulated copper wires surrounded by an outer protective jacket. If the cable has an outer foil shielding, it is referred to as *shielded twisted pair (STP)*, which adds protection from radio frequency interference (RFI) and EMI. Twisted-pair cabling, which does not have this extra outer shielding, is called *unshielded twisted pair (UTP)*.

The twisted-pair cable contains copper wires that twist around each other, as shown in Figure 14-3. This twisting of the wires protects the integrity and strength of the signals they carry. Each wire forms a balanced circuit, because the voltage in each pair uses the same amplitude, just with opposite phases. The tighter the twisting of the wires, the more

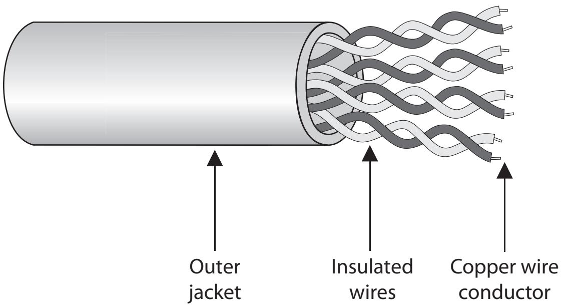


Figure 14-3
Twisted-pair cabling uses copper wires.

UTP Category	Characteristics	Usage
Category 1	Voice-grade telephone cable for up to 1 Mbps transmission rate	No longer in use for data or phones.
Category 2	Data transmission up to 4 Mbps	Historically used in mainframe and minicomputer terminal connections, but no longer in common use.
Category 3	10 Mbps for Ethernet	Used in older 10Base-T network installations and legacy phone lines.
Category 4	16 Mbps	Normally used in Token Ring networks.
Category 5	100 Mbps; two twisted pairs	Sometimes used in legacy 100Base-TX; deprecated in 2001 for data but still used for telephone and video.
Category 5e	1 Gbps; four twisted pairs, providing reduced crosstalk	Widely used in modern networks.
Category 6	1 Gbps, but can support 10 Gbps up to 55 meters	Used in newer network installations requiring high-speed transmission. Standard for Gigabit Ethernet.

Table 14-2 UTP Cable Ratings

resistant the cable is to interference and attenuation. UTP has several categories of cabling, each of which has its own unique characteristics.

The twisting of the wires, the type of insulation used, the quality of the conductive material, and the shielding of the wire determine the rate at which data can be transmitted. The UTP ratings indicate which of these components were used when the cables were manufactured. Some types are more suitable and effective for specific uses and environments. Table 14-2 lists the cable ratings.

Copper cable has been around for many years. It is inexpensive and easy to use. A majority of the telephone systems today use copper cabling with the rating of voice grade. Twisted-pair wiring is the preferred network cabling, but it also has its drawbacks. Copper actually resists the flow of electrons, which causes a signal to degrade after it has traveled a certain distance. This is why cable lengths are recommended for copper cables; if these recommendations are not followed, a network could experience signal loss and data corruption. Copper also radiates energy, which means information can be monitored and captured by intruders. UTP is the least secure networking cable compared to coaxial and fiber. If an organization requires higher speed, higher security, and cables to have longer runs than what is allowed in copper cabling, fiber-optic cable may be a better choice.

Fiber-Optic Cable

Twisted-pair cable and coaxial cable use copper wires as their data transmission media, but fiber-optic cable uses a type of glass that carries light waves, onto which we modulate the data being transmitted. The glass core is surrounded by a protective cladding, which in turn is encased within an outer jacket.

Fiber Components

Fiber-optic cables are made up of a light source, an optical fiber cable, and a light detector.

Light Sources Convert electrical signal into light signal.

- Light-emitting diodes (LEDs)
- Diode lasers

Optical Fiber Cable Data travels as light.

- **Single mode** Small glass core, used for high-speed data transmission over long distances. They are less susceptible to attenuation than multimode fibers.
- **Multimode** Large glass core, able to carry more data than single mode fibers, though they are best for shorter distances because of their higher attenuation levels.

Light Detector Converts light signal back into electrical signal.

Because it uses glass, *fiber-optic* cabling has higher transmission speeds that allow signals to travel over longer distances. Fiber-optic cabling is not as affected by attenuation and EMI when compared to cabling that uses copper. It does not radiate signals, as does UTP cabling, and is difficult to eavesdrop on; therefore, fiber-optic cabling is much more secure than UTP, STP, or coaxial.

Using fiber-optic cable sounds like the way to go, so you might wonder why you would even bother with UTP, STP, or coaxial. Unfortunately, fiber-optic cable is expensive and difficult to work with. It is usually used in backbone networks and environments that require high data transfer rates. Most networks use UTP and connect to a backbone that uses fiber.

NOTE The price of fiber and the cost of installation have been steadily decreasing, while the demand for more bandwidth only increases. More organizations and service providers are installing fiber directly to the end user.

Cabling Problems

Cables are extremely important within networks, and when they experience problems, the whole network could experience problems. This section addresses some of the more common cabling issues many networks experience.

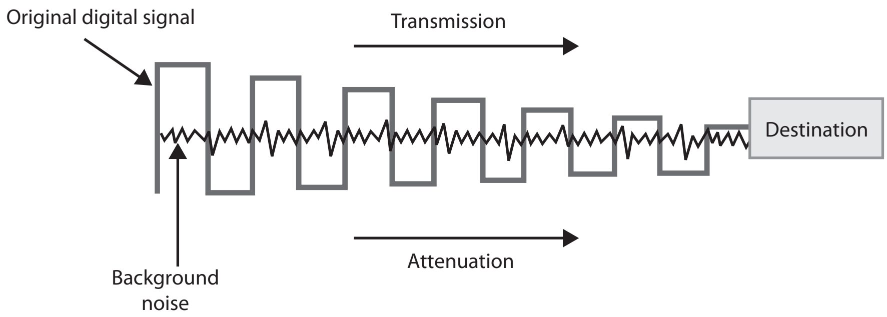


Figure 14-4 Background noise can merge with an electronic signal and alter the signal's integrity.

Noise The term *line noise* refers to random fluctuations in electrical-magnetic impulses that are carried along a physical medium. Noise on a line is usually caused by surrounding devices or by characteristics of the wiring's environment. Noise can be caused by motors, computers, copy machines, fluorescent lighting, and microwave ovens, to name a few. This background noise can combine with the data being transmitted over the cable and distort the signal, as shown in Figure 14-4. The more noise there is interacting with the cable, the more likely the receiving end will not receive the data in the form originally transmitted.

Attenuation *Attenuation* is the loss of signal strength as it travels. This is akin to rolling a ball down the floor; as it travels, air causes resistance that slows it down and eventually stops it. In the case of electricity, the metal in the wire also offers resistance to the flow of electricity. Though some materials such as copper and gold offer very little resistance, it is still there. The longer a wire, the more attenuation occurs, which causes the signal carrying the data to deteriorate. This is why standards include suggested cable-run lengths.

The effects of attenuation increase with higher frequencies; thus, 100Base-TX at 80 MHz has a higher attenuation rate than 10Base-T at 10 MHz. This means that cables used to transmit data at higher frequencies should have shorter cable runs to ensure attenuation does not become an issue.

If a networking cable is too long, attenuation will become a problem. Basically, the data is in the form of electrons, and these electrons have to "swim" through a copper wire. However, this is more like swimming upstream, because there is a lot of resistance on the electrons working in this media. After a certain distance, the electrons start to slow down and their encoding format loses form. If the form gets too degraded, the receiving system cannot interpret the electrons any longer. If a network administrator needs to run a cable longer than its recommended segment length, she needs to insert a repeater or some type of device that amplifies the signal and ensures that it gets to its destination in the right encoding format.

Attenuation can also be caused by cable breaks and malfunctions. This is why cables should be tested. If a cable is suspected of attenuation problems, cable testers can inject signals into the cable and read the results at the end of the cable.