CHAPTER 6

Securing TCP/IP

After reading this chapter, you will be able to:

- Explain the role that the Transmission Control Protocol (TCP) and the Internet Protocol (IP) play in computer networking
- Understand how security concepts integrate into the OSI networking models
- Identify the major components of the TCP/IP protocol suite and explain how each is used in networking
- Decipher the contents of a TCP/IP packet and describe the types of modifications involved in malformed packet attacks
- Describe the enhancements provided by adding IPSec security to a network
- Identify the various security protocols used to enhance Web communications and choose the protocol appropriate for a given situation

The vast majority of computer networks, including the Internet itself, are dependent upon a set of protocols known as the TCP/IP suite. The two core components of this suite, the Transmission Control Protocol (TCP) and the Internet Protocol (IP), control the formatting and routing of data as it flows from point to point across the network. Although a large number of other network protocols are in use today (such as Novell's Internetwork Packet Exchange/Sequenced Packet Exchange [IPX/SPX] and Apple's AppleTalk), the discussion in this book is limited to these popular protocols because they are the "language of the Internet" and the source of many security vulnerabilities.

6.1 Introduction to Transmission Control Protocol/ Internet Protocol (TCP/IP)

Although the TCP/IP suite has been modified and enhanced over the years, the core set of protocols date back to the earliest days of the Internet, when it was a private network interconnecting several large U.S. government research sites. These protocols completely describe the ways that devices communicate on TCP/IP networks, ranging all the way from the way individual chunks of data (known as **packets**) are formatted to the details of how those packets are routed through various networks to their final destinations.

In this section, we introduce the basic concepts behind the TCP/IP suite. You'll first learn about the four protocols that form the basic building blocks of TCP/IP. Next, you'll learn about how the Open Systems Interconnection (OSI) reference model governs the design of TCP/IP and other networking protocols. Finally, you'll learn how to examine the "guts" of a packet and actually interpret those electrical impulses as they transit a network.

6.1.1 TCP/IP Protocols

Four main protocols form the core of TCP/IP: the Internet Protocol (IP), the Transmission Control Protocol (TCP), the User Datagram Protocol (UDP), and the Internet Control Message Protocol (ICMP). These protocols are essential components that must be supported by every device that communicates on a TCP/IP network. Each serves a distinct purpose and is worthy of further discussion.

Internet Protocol

The **Internet Protocol (IP)** is a network protocol that provides essential routing functions for all packets transiting a TCP/IP network. By this point in your computer science education, you're probably familiar with the concept of how IP addresses uniquely identify network destinations. Each system connected to the Internet and available for public use is assigned an IP address that allows other systems to locate it on the global network. (There are some exceptions that you'll learn about later in this book. Sometimes multiple systems share a single IP address for security and/or efficiency reasons using a service known as Network Address Translation [NAT].)

The Internet Protocol provides networking devices (workstations, servers, routers, switches, and so on) with guidance on how to handle incoming packets. Each IP datagram bears a source IP address that identifies the sender and a destination IP address that identifies the recipient. When a device receives an IP datagram, it first checks to see whether the destination IP address is an IP address assigned to the local machine. If it is, it processes the datagram locally. If not, it determines the proper place to forward the packet (the "next hop") to help it along toward its ultimate destination. IP is responsible for ensuring that systems can identify the next hop in an efficient manner so that all network traffic eventually reaches its ultimate destination.

It's important to note that the IP protocol itself does not provide any reliability guarantees; that is, IP provides no assurance to users that a packet will reach its ultimate destination. This is the responsibility of other protocols within the TCP/IP suite.

Besides addressing, the other main responsibility of IP is datagram **fragmentation**. As a datagram travels from source to destination, it may pass through several intermediate networks with varying topologies. Each of those networks may specify a different maximum datagram size. Because the originating machine has no way of telling what networks a datagram will pass through, let alone the maximum datagram size on those networks, IP must accommodate those limits in a method transparent to the end users. This is where fragmentation comes into play. If a datagram reaching a network exceeds the maximum length permissible for that network, IP breaks the datagram up into two or more fragments, each of which complies with the maximum length for that network. Each fragment is labeled

NOTE

Throughout this section, you'll see individual units of data referred to as either IP datagrams or TCP packets. Many people use these terms interchangeably, but that is not technically correct. IP and UDP work with datagrams, whereas TCP processes packets (sometimes referred to as segments).

NOTE

The material on IP routing presented in this book is intended to be a brief refresher only. We have assumed that students have a familiarity with basic networking, routing, addressing, and network devices. If this is not the case, please take the time to review this material in a networking text.

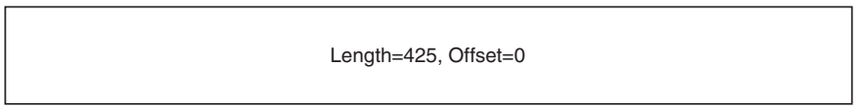
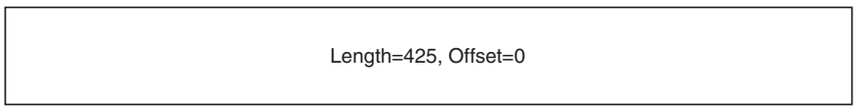


Figure 6.1
Original datagram



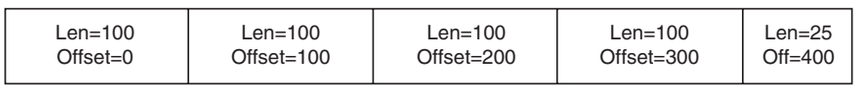


Figure 6.2
Fragmented datagram

with a length and an offset (both specified in bytes). The length simply specifies the total number of bytes in the fragment. The offset specifies the location of the first byte of the fragment in the original datagram. Therefore, the first fragment always has an offset of 0.

When a host wishes to reassemble a fragmented datagram, it merely puts all the pieces together using the length and offset information. This process is best understood through the use of an example. Imagine a network where the maximum datagram length is 100 bytes. Assume that the network receives an inbound datagram that is 425 bytes in length, as shown in Figure 6.1.

TIP

A thorough understanding of IP fragmentation is essential to understanding several networking vulnerabilities presented later in this chapter. Take the time to ensure that you comprehend this material.

It would be impossible for that network to carry the original datagram, so IP breaks it up into five fragments, as shown in Figure 6.2.

Take a moment to ensure that you understand how IP derived the length and offset of each fragment. The first four fragments each contains 100 bytes of data, so they all have a length of 100. The final fragment contains the remaining 25 bytes of data, so it has a length of 25. The first fragment always has an offset of 0 and, therefore, occupies bytes 0–99 of the original datagram. The second fragment then occupies bytes 100–199, so it has an offset of 100, indicating that the first byte of the second fragment should be placed in the 100th byte of the reassembled datagram. The third, fourth, and fifth datagrams similarly receive offsets of 200, 300, and 400, respectively.

Transmission Control Protocol

The **Transmission Control Protocol (TCP)** provides networks with a reliable mechanism for process-to-process communication. TCP "rides" on top of the Internet Protocol. After a TCP packet is constructed, it is transformed into an IP datagram by adding appropriate addressing and fragmentation information. This process, known as **encapsulation**, is discussed later in this chapter. There are three critical features of TCP:

- TCP is a *reliable* protocol that guarantees delivery of individual packets from the source to the destination. This is accomplished through the use of an acknowledgement system where the receiving system informs the sender that it has received the packet successfully. TCP's reliability is sometimes compared to the reliability of sending a "return receipt requested" letter through the mail. The sender receives a confirmation notice when the recipient receives the original message.
- TCP provides *error-checking*. Each packet contains a checksum that the recipient uses to ensure that the data was not corrupted while in transit. If the checksum does not match the data, the receiver asks the sender to retransmit the packet. From a security perspective, it's important to note that TCP's error-checking functionality does not provide any security against malicious tampering; it merely ensures that the data was not corrupted accidentally while in transit.
- TCP is *connection-oriented*. It uses a session establishment and teardown algorithm that creates dedicated channels of communication between two processes.

Each TCP connection begins with the use of a three-way handshaking process that establishes a two-way communications channel between two processes. This handshaking process, shown in Figure 6.3, takes advantage of two binary fields—the SYN and ACK flags. When a process on one machine wishes to establish a connection with a process on another machine, it sends a single packet with the SYN flag set to signify a connection request. If the destination host wishes to establish the communication, it acknowledges the opening of a communications channel from the source to the destination by replying with a packet that has the ACK flag set. It uses this same packet to set up a communications channel from the original

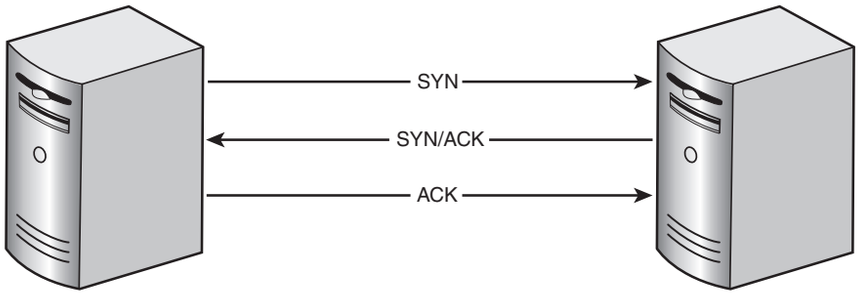


Figure 6.3
Three-way TCP handshake

source to the original destination by setting the SYN flag of that packet. When the original source receives the "SYN-ACK" packet, the first channel of communication is open. It then sends a third packet with the ACK flag set to complete the handshaking process by acknowledging the opening of the second channel of communications.

Throughout our discussion of TCP, we've referred to the fact that it is used to establish connections between *processes* on two systems. TCP allows the unique addressing of a process (such as a Web server or a mail transfer program) through the use of **ports**. Each port uniquely identifies a particular process on a system and, when combined with an IP address, uniquely identifies the combination of a system and process often referred to as a **socket**.

NOTE

It's important to note that although many ports are commonly associated with specific protocols (for example, Web servers ordinarily run on port 80 whereas SMTP servers run on port 25), these are not *required* associations. You can configure a Web server to run on port 8080, 800, 25 or any other port. Malicious individuals sometimes take advantage of this fact to hide the true nature of network traffic.

User Datagram Protocol

The **User Datagram Protocol (UDP)** is a companion to TCP. Like its counterpart, it is a transport protocol that rides on top of the Internet Protocol. Unlike TCP, UDP is a connectionless protocol that does not provide the reliability of guaranteed datagram delivery. It merely makes a best effort to deliver a packet from one process to another. It is up to higher level software to provide reliability, if desired.

The major advantage to UDP is the extremely low overhead involved in connectionless communications. UDP datagrams may be sent freely between hosts without the lengthy three-way handshaking process required by TCP. Additionally, UDP does not need to keep track of the sequencing and acknowledgement information that TCP uses to provide reliable packet delivery. UDP is often used for applications such as streaming media that do not depend on the guaranteed delivery of every packet. You may recall from our discussion of TCP that we compared TCP's reliability to sending a return receipt requested letter. UDP is analogous to putting on a first-class stamp, dropping the letter in a mailbox, and hoping for the best.

REQUEST FOR COMMENTS

The open nature of the Internet requires that every participant follow the same sets of standards when communicating with other hosts. If this were not the case, communicating systems would have no way of deciphering the bits and bytes sent and received across the network wire. Hardware and software developers, therefore, follow a common set of standards when creating systems that communicate using TCP/IP. These standards are written up in documents known as Requests for Comments (RFCs) and are published by the Internet Engineering Task Force (IETF).

These documents are full of detailed technical specifications, but are often written in a surprisingly clear manner. If you ever find yourself attempting to analyze packets at the bit level and become confused, you may wish to utilize one or more of these documents. You can find them at a large number of sites on the Internet, including the central IETF repository located at *http://www.ietf.org/rfc.html*. Each RFC document pertains to a specific protocol or application, and has a unique number for easy identification. For your quick reference, some commonly used RFCs include:

- **RFC 768:** User Datagram Protocol
- **RFC 791:** Internet Protocol
- **RFC 792:** Internet Control Message Protocol
- **RFC 793:** Transmission Control Protocol
- **RFC 2821:** Simple Mail Transfer Protocol

Internet Control Message Protocol

The final protocol we will examine, the **Internet Control Message Protocol (ICMP)** is the administrative workhorse of the TCP/IP suite. It is responsible for transmitting control messages between networked devices. Unlike TCP and UDP, ICMP does not use IP *per se* to provide datagram delivery; however, it does incorporate basic portions of the IP header so that it can use the same routing infrastructure as IP. Figure 6.4 illustrates the relationships among TCP, IP, UDP, and ICMP.

ICMP is used to deliver many different types of administrative control messages. Some examples are:

- Network/host/port unreachable
- Packet time to live expired

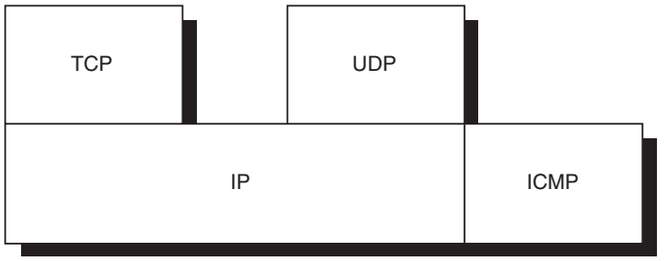


Figure 6.4
Relationships among protocols in the TCP/IP suite

- Source quench (used when a gateway is overloaded and wishes to pause incoming traffic)
- Redirect messages (used to reroute traffic)
- Echo request and echo reply messages (used to determine whether a host is active on the network; these messages are used by the **ping** command)

6.1.2 Open Systems Interconnection Model

The Open Systems Interconnection (OSI) reference model, shown in Figure 6.5, was developed by the International Organization for Standardization in the late 1970s in an effort to describe the basic functionality of networked data communications. The model consists of seven layers: Application layer, Presentation layer, Session layer, Transport layer, Network layer, Data Link layer, and Physical layer.

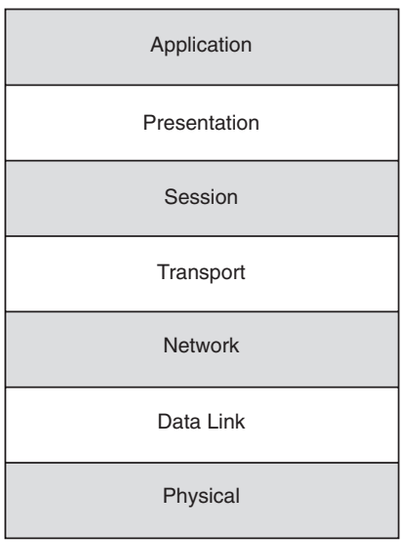


Figure 6.5
OSI model

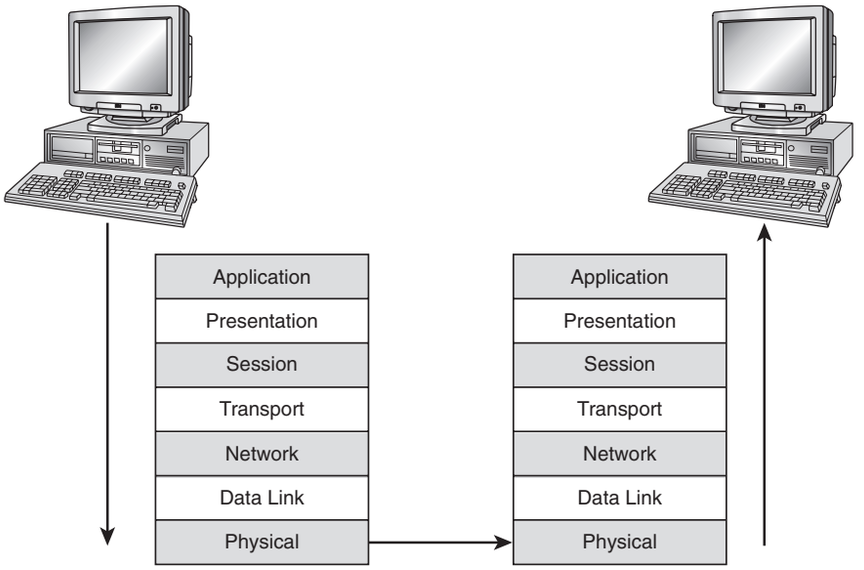


Figure 6.6
Encapsulation using the OSI model

The OSI model uses a process known as encapsulation to sequentially process data through the various model layers until it is ready for transmission across a network medium (e.g., a copper wire or fiber-optic cable). Each layer of the OSI model performs some transformation of the data, either by adding a header that encapsulates the data received from the previous layer or by converting the data into another form (such as from binary data into electrical impulses). When the remote device receives the packet, it also processes it through the layers of the OSI model, but in reverse order. At the conclusion of the process, the destination machine has the same data that was sent by the originating machine. The process of encapsulation is illustrated in Figure 6.6.

The beauty of the OSI model lies in the ability of system developers to take advantage of abstraction. A programmer writing software that works at the Application layer doesn't need to worry about the details of how the lower layers work. If the software communicates with other systems, the programmer may simply view it as communication between the systems at the Application layer. The encapsulation process ensures that the networking protocols take care of the other details. The OSI model is a fundamental principle of networking, and many texts devote entire chapters to fully exploring the model. Keeping with the focus of this text, we'll briefly describe each layer and provide information on its relevance to information security practitioners.

Application Layer

The Application layer is the highest layer encountered in the OSI model. It consists of the software that directly interacts with computer users and provides the standards that govern how those users manipulate the system. The Application layer is home to innumerable applications, including electronic mail software, Web browsers, office productivity suites, financial tools, and other packages.

The vast majority of security vulnerabilities inherent in computing systems occur at the Application layer of the OSI model. This layer includes almost all malicious code objects, such as viruses, worms, and Trojan horses. The exploits that take place at the Application layer are discussed in detail in Chapter 7.

Presentation Layer

The Presentation layer is responsible for taking data from the lower layers and converting it into a format usable by the Application layer. This layer is responsible for taking the formats (both proprietary and standardized) used by various applications and allowing the data within those applications to be shared. Common standards found at the Presentation layer include the ASCII, ANSI, and EBCDIC character sets.

The most important activity that takes place at the Presentation layer from a security perspective is encryption. This layer is responsible for making encryption and decryption of data transmitted over the network transparent to the end user. It handles these mathematical processes to ensure that the end user receives a secure and efficient computing experience.

Session Layer

The Session layer is responsible for the creation, teardown, and maintenance of network connections between processes that are used with connection-oriented protocols such as TCP. It is the location of the three-way handshaking process that takes place to establish TCP communications between two hosts.

A common exploit malicious individuals use to take advantage of security vulnerabilities at this layer is session hijacking. With most unencrypted application traffic, authentication takes place at the beginning of a communications sequence. Consider the case of a typical Telnet session—you ini-

tiate communications with a remote host and provide a username and password to prove your identity to the remote system. Once you've completed that process, you are never again required to authenticate yourself. A malicious individual may try to use a session hijacking attack to "take over" your session by disabling your computer in the midst of the communication and responding to the other system as if it were you.

Transport Layer

The Transport layer is responsible for managing the flow of data between two systems. It includes error recovery functionality, message acknowledgments, and flow control mechanisms. Two of the most common transport protocols that operate at this layer are TCP and UDP.

A large number of vulnerabilities are present at this layer of the OSI model. One attack found at the Transport layer is the SYN Flood attack, which takes advantage of weaknesses in the way some operating systems handle TCP's three-way handshaking process. This attack is described in further detail in the next chapter.

Other Transport-layer attacks exploit buffer overflow vulnerabilities in various components of the TCP/IP stack. Buffer overflows are the result of poor programming practices that allow a cleverly crafted series of actions to cause the overfilling of a memory space reserved for a certain task. The results of a buffer overflow range from the annoying (the system restarts) to the severe (the perpetrator of the attack gains complete administrative control of the targeted system).

Network Layer

In the TCP the Network layer IP model, consists of the Internet Protocol. This layer is responsible for ensuring that datagrams are properly routed across the network from their source to their ultimate destination. The mapping of physical to logical addresses and fragmentation of datagrams take place at this layer of the OSI model.

Several classes of attacks exploit weaknesses in the Network layer. Fragmentation attacks are one such class. Recall from earlier in the chapter that IP is capable of taking datagrams that are too large for a particular network and dividing them up into smaller fragments suitable for transmission. Malicious individuals can wreak havoc on unprepared systems by cleverly manipulating datagram fragments so that one of two conditions occurs:

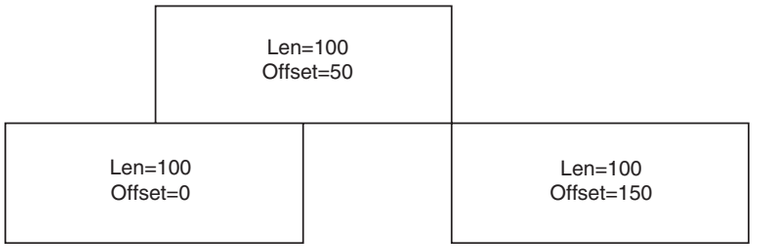


Figure 6.7
Overlapping fragment attack

- **Two fragments overlap:** This takes place when the offset field for one fragment contains a value that places it within the space occupied by the previous fragment. An example of an overlapping fragment attack is shown in Figure 6.7.
- **Two adjacent fragments do not meet:** This occurs when one fragment contains an offset that causes a gap between it and the immediately preceding fragment. An example is shown in Figure 6.8.

Both of these attacks have been around for quite some time and are well known to security administrators and operating system developers. Most modern implementations of the TCP/IP stack are immune to these vulnerabilities, but many computers on the Internet running older operating systems may still be vulnerable, if appropriate security patches have not been applied.

Data Link Layer

The Data Link layer is responsible for the conversion between upper layer datagrams and the digital language of bits—the 1s and 0s capable of transmission across a computer network. The two sublayers of the Data Link layer—the Logical Link Control (LLC) sublayer and the Media Access Control (MAC) sublayer—each have distinct functions. The LLC sublayer is responsible for three functions:

- Error correction

Figure 6.8
Nonadjacent fragment attack

- Flow control
- Frame synchronization

Most network administrators are more familiar with the MAC sublayer because it shares a name with the physical addressing scheme used by computer networks—the MAC addressing scheme. These low-level addresses are completely independent of the upper layers. In fact, they are actually assigned to each networking device at the factory and are normally never changed during the device's lifetime. The 48-bit (12-byte) address has two components: the first six hexadecimal characters identify the manufacturer of the networking device and are selected by the manufacturer from a range assigned to it by the Institute of Electrical and Electronics Engineers (IEEE). The last six hexadecimal characters uniquely identify the device and are normally assigned in a sequential manner by the manufacturer. Under normal circumstances, each MAC address should be globally unique; that is, no two devices anywhere in the world should share the same MAC address. Of course, this is dependent upon the ability of each manufacturer to correctly manage its assigned address space. Consider the MAC address 00:00:0C:45:12:A6. The first six characters (00:00:0C) identify the manufacturer (in this case, Cisco Systems) whereas the last six characters (45:12:A6) uniquely identify the device.

TIP

For a complete listing of IEEE-assigned manufacturer IDs, visit *http://standards.ieee.org/regauth/oui/oui.txt*.

During the conversion between the Network layer and the Data Link layer, networking devices make use of the Address Resolution Protocol (ARP). This protocol provides the capability of determining a remote device's physical address by polling a network using its IP address. A similar protocol, the Reverse Address Resolution Protocol (RARP), handles conversions between MAC addresses and IP addresses when necessary. Every time a packet reaches a new intermediate device, that device determines the appropriate "next hop" address using ARP or an internal MAC address cache, and changes the destination MAC address of the packet accordingly.

Physical Layer

The lowest layer of the OSI model is the Physical layer. This layer converts the Data Link layer's data bits into the actual impulses that are transmitted over the physical network. The type of impulse is dependent upon the type of media used on the network. For example, in a network that uses twisted-pair or coaxial cabling with copper conductors, the Physical layer will convert bits

into electrical impulses. On the other hand, the same bits will be converted into pulses of light on network segments utilizing fiber-optic cabling.

The Physical layer is subject to a new class of threats that does not affect the other layers: physical threats. If a hacker has access to a physical component of a computer network, such as a computer attached to that network or the cables themselves, he or she may be able to use a hardware or software packet sniffer to monitor traffic on that network. These devices simply monitor the network wire and capture the bits that pass, allowing the user to monitor all traffic on the network. If these packets are unencrypted, the hacker essentially has full access to view (and possibly modify) all network traffic, effectively invalidating most of the security controls put in place by administrators.

6.2 Anatomy of a Packet

In the last section, you learned how hackers might use a packet sniffer to monitor traffic on a computer network. These devices are also useful to administrators seeking to hunt down the source of malicious activity or troubleshoot networking issues. They allow the user to see details of each packet that are not normally displayed by a typical computer system.

TIP

Several free packet-sniffing utilities are available on the Internet. Consider evaluating programs such as tcpdump (for UNIX) or windump (for Windows). You can locate downloadable versions of these packages through an Internet search. More sophisticated packet sniffers (including hardware sniffers) are available from a number of commercial sources.

To effectively use these tools, it's important that you understand the components of a packet and their formatting and structure. Each packet has two main components: a payload and a header. The packet payload is the "meat" of the packet—it's the actual data that one system wishes to transmit to another. The header is all of the protocol and routing information necessary to facilitate the transmission of the packet over the network. As a packet travels down the OSI model during the encapsulation process, each layer adds information to the front of the packet header. As the packet traverses the network, each device processes as much of the packet header as is necessary to carry out its function. For example, a switch may need only to read and modify the Physical and Data Link layer headers to modify a packet's destination MAC address, whereas a router reaches down to the Network layer to read the IP address of the packet's final destination.

When you use a packet sniffer to analyze packets, you have the option of displaying the data in a number of different forms. The sniffer should be capable of displaying the actual binary (0/1) bits of the packet or converting it into a number of other forms, including hexadecimal, ASCII, and

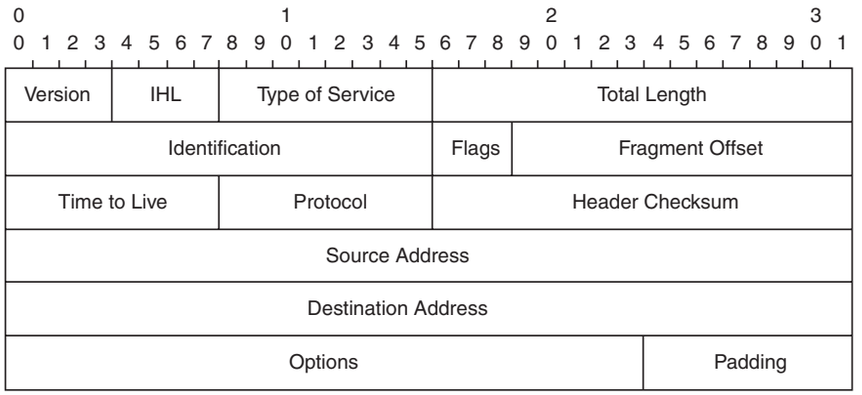


Figure 6.9
IP header (source: RFC 791)

ANSI. You should use whatever method you find provides the easiest results for you to interpret.

6.2.1 Packet Header

As mentioned in the previous section, packet headers are built sequentially, with additional information added at each layer of the encapsulation process. Let's take some time to look at the header information added by three common protocols: IP, TCP, and UDP.

IP Header

The IP header, shown in Figure 6.9, contains all of the information needed by the Internet Protocol to carry out its routing functions. Figure 6.9 is taken directly from the IP specification found in RFC 791. Examine the header carefully and familiarize yourself with its contents. If you're unclear about any of the specific fields, you may wish to consult RFC 791 for further details. You should have at least a basic familiarity with the following:

- The *Total Length* and *Offset* fields are important in datagram fragmentation and reassembly.
- The *Protocol* field identifies the higher-level transport protocol being used by the datagram. Values commonly found here include 1 for ICMP, 6 for TCP, and 11 for UDP.
- The *Source Address* and *Destination Address* contain the IP addresses of the communicating hosts.

NOTE

Many students find analysis of packet headers a tedious and confusing process. Although it is clearly not the most exciting part of the information security profession, it is extremely important. A large number of security exploits (such as fragmentation attacks) are difficult, if not impossible, to detect through analysis at higher levels of processing and may only be identified through the use of packet analysis.

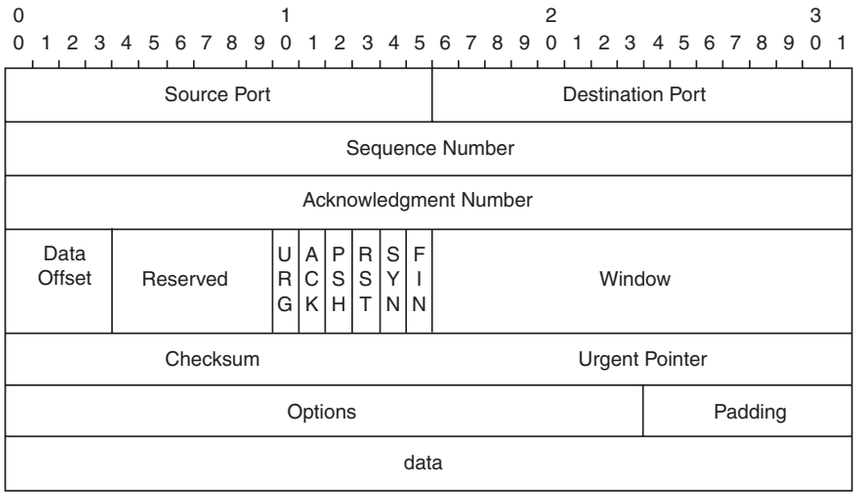


Figure 6.10
TCP header (source: RFC 793)

TCP Header

If the IP header information identifies the packet as a TCP packet, you may further dissect it by analyzing the TCP header. The header format, as specified by RFC 793, is shown in Figure 6.10.

You'll notice that all of the fields contained within the TCP header are directly related to the services provided by TCP. You won't find IP addresses or lower-level routing information here; those are handled by IP. You will find fields that assist with the error correction, sequencing, and connection creation/maintenance/tear-down process. Some important details you may wish to take note of include:

- The *Source Port* and *Destination Port* are used to uniquely identify the processes that are communicating with each other. Some values commonly found here are shown in Table 6.1.
- The *Sequence Number* and *Acknowledgement Number* are used by TCP to place packets in the proper order and ensure that they reach their final destination.
- The *SYN* and *ACK* flags are used in the three-way handshaking process described earlier in this chapter that is used to create a connection.
- The *RST* and *FIN* flags are used in a similar three-way handshaking process that tears down a connection. (The host wishing to close

TABLE 6.1 Well-Known Ports

Well-Known Port	Service
20/21	FTP
23	Telnet
25	SMTP
53	DNS
80	HTTP
110	POP
443	SSL

the connection sends a single RST packet to request closure of one side of the communication. The second host sends a packet with both the FIN and RST flags set that completes the first RST request and initiates the second. The initiating host then sends a FIN packet to complete the connection closure.)

- The *checksum* is used to ensure that the data received by the destination host is identical to the data that was sent by the sender. Note that this field provides error correction, not security. If an intermediate party intercepts an unencrypted TCP packet, he or she can alter both the packet payload and the checksum and defeat the error correction process.

UDP Header

As you would expect, the UDP protocol also adds header information to a packet when it is the transport protocol of choice. UDP is a connectionless protocol that provides fewer services than TCP, so the header is somewhat shorter (it only contains four fields). Figure 6.11 contains the UDP header specification found in RFC 768.

The fields of the UDP header are as follows:

- The *Source Port* and *Destination Port* are used to identify the communicating processes.

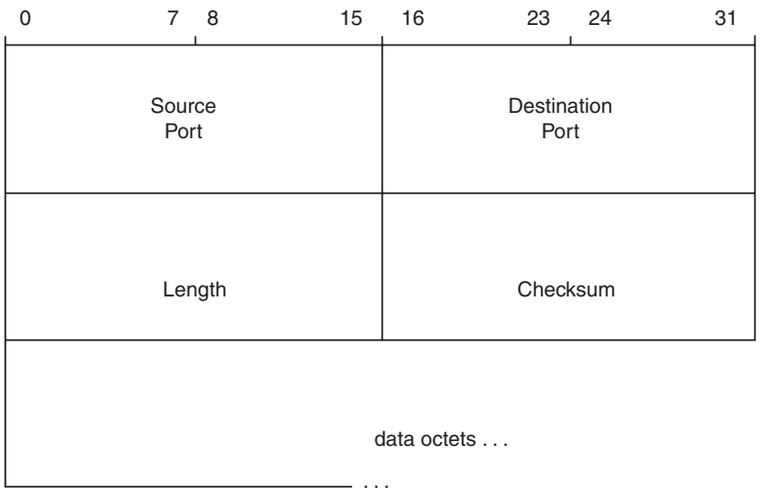


Figure 6.11
UDP header (source: RFC 768)

- The *Length* field contains the number of bytes in the datagram (including both the data payload and the datagram header).
- The *Checksum* field is used for error correction in the same manner as the TCP checksum.

6.2.2 Packet Payload

It's easy to get wrapped up in header analysis and forget the purpose of a packet: to carry data from one point on a network to another. The payload of the packet usually comprises the great majority of the data within the packet and contains the actual data that is being exchanged as part of the communication between the two systems. The packet payload can contain any type of data that can be expressed in binary form. Once you've analyzed a large number of packets, you may begin to notice telltale signs of packet payloads common on your network. For example, you may start to recognize the binary string that precedes a file formatted using an image type commonly found on your network.

6.3 Internet Protocol Security (IPSec)

As you no doubt have realized by this point in the chapter, TCP/IP is fraught with insecurities. This is inherent in its nature. Keep in mind that

the TCP/IP suite was designed to operate on a government network with a relatively small number of hosts who completely trusted each other. Security was simply not a concern to the designers of the Internet. Now that the network has expanded to global use and large numbers of people around the world are connected, security is paramount.

Security developers have engineered a number of solutions designed to add security to this inherently insecure infrastructure. One of the major developments of the past decade is the launch of **IPSec**, a security-enhanced version of the Internet Protocol. IPSec is designed to add several different layers of security to the communications process and work in a manner that is transparent to the end user. Support is optional in systems that implement IPv4, but IPSec support is required for systems that implement IPv6.

6.3.1 Protocols

IPSec makes use of three major protocols to provide security services to communicating systems:

- All IPSec communication is based on the establishment of security associations (SAs) between communicating systems. These SAs contain the identification and keying material used to create the secure session. The **Internet Security Association and Key Management Protocol (ISAKMP)** is responsible for the creation and maintenance of SAs in an IPSec environment.
- The **Authentication Header (AH)** adds information to the header of a packet to provide two critical cryptographic functions: integrity and authentication. When AH is used in an IPSec environment, systems making use of this service to communicate with each other can be assured that the communications they receive were actually sent by the purported sender and were not altered while in transit. It is important to note that the use of AH alone does *not* provide any confidentiality services.
- The **Encapsulating Security Payload (ESP)** adds guarantees of confidentiality. Like AH, it provides integrity and authentication, but it also adds data encryption to ensure that the contents of packets are kept safe from eavesdroppers. As you'll learn in the next

NOTE

There is a fourth protocol, the IP Payload Compression (IPcomp) protocol, which increases IPSec performance through the use of compression technology.

section, the degree of confidentiality provided by ESP depends on the mode in which IPSec is run.

6.3.2 Encryption Modes

IPSec's ESP supports two different modes of operation. Each mode provides different services, and its confidentiality guarantee covers different subsets of packet data. Each mode is suitable for different networking circumstances.

- **Transport mode** is used mainly for host-to-host communication over a network that may or may not support IPSec. When run in this mode, ESP provides confidentiality for the data contained within the payload. However, the packet header data must remain unencrypted so that intermediate hosts not participating in the IPSec security association can properly forward it. If the data were encrypted, intermediate hosts would not be able to read the addressing information and would be unable to determine the appropriate action to take with the packet.
- **Tunnel mode** creates virtual circuits between two networking devices and encrypts all of the data passed between them. Because virtual circuits are used, ESP in tunnel mode can also encrypt the packet header. This added security prevents traffic analysis attacks, where a malicious individual monitors a network and, despite the fact that he or she cannot read the encrypted packet payloads, may glean useful information simply from determining which hosts are communicating with each other and the frequency at which they communicate. Tunnel mode is often found in gateway-to-gateway communications, such as two firewalls communicating with each other. When packets reach the destination gateway, they are decrypted and processed appropriately.

6.4 Web Security

The World Wide Web comprises a large portion of the traffic on today's Internet, second only to electronic mail. In Chapter 5, you learned how security may be added to email communications through the use of cryp-

tographic technologies. In this section, we examine two common technologies used to add security to Web communications: SSL and HTTP-S.

6.4.1 Secure Sockets Layer (SSL)

Netscape developed the **Secure Sockets Layer (SSL)** to provide confidentiality, integrity, and authentication to any Application-layer protocol that supports SSL. However, its use is mainly seen in the securing of communications between Web browser clients and Web servers. When used in this scenario, it is known as HTTP over SSL, and is abbreviated as "https." This is how the common secure Web site prefix https:// is derived. SSL is evolving into the new Transport Layer Security (TLS) standard. More information on TLS is available in RFC 2246 (online at *http://www.ietf.org/rfc/rfc2246.txt*).

SSL works by facilitating the exchange of digital certificates between systems. A required element of SSL is that the server must send a digital certificate to the client to provide a public key and verify its identity. After the client is satisfied as to the authenticity of the certificate (through an automated process conducted by the browser software), the two hosts may begin communicating using encrypted communications. Optionally, SSL allows the client to provide a digital certificate to the user in order to prove the client's identity. However, this optional component of SSL is rarely used, because very few individuals have their own SSL certificates.

6.4.2 Secure-HTTP (HTTP-S)

Secure-HTTP (HTTP-S) offers an alternative to SSL for those seeking to secure Web connections. Like SSL, HTTP-S provides confidentiality, integrity, and authentication services. However, there are several key differences:

- SSL is a connection-oriented protocol (designed to support sessions), whereas HTTP-S is a connectionless protocol (designed to facilitate the transmission of individual messages).
- SSL is embedded in most popular Web browsers, whereas HTTP-S is found only in a few less common browsers.
- SSL functions between the Session and Transport layers, whereas HTTP-S functions at the Application layer.

WARNING

The terminology used to describe secure Web protocols is confusing. The Secure Sockets Layer (SSL) is often referred to as HTTP over SSL and abbreviated "https." The Secure-HTTP protocol, on the other hand, is often abbreviated "HTTP-S." Be sure you understand the difference!

6.5 Chapter Summary

- The Internet Protocol is responsible for providing essential routing functions for all traffic on a TCP/IP network.
- The Transmission Control Protocol and User Datagram Protocol are the most commonly used transport protocols on TCP/IP networks. TCP provides connection-oriented communication, whereas UDP provides connectionless communications.
- TCP connections are initiated and terminated with a three-way handshaking process.
- The Internet Control Message Protocol provides administrative services to TCP/IP networks.
- The seven-layer OSI model provides a general framework for the transmission of data across a network. The layers (from highest to lowest) are Application, Presentation, Session, Transport, Network, Data Link, and Physical.
- Packets consist of a header and a payload. The header is created as a result of the encapsulation process and contains specific information for each protocol used in the communication. The payload contains the actual data being transmitted between hosts.
- IPSec provides security enhancements for the TCP/IP suite. The major IPSec protocols are the Authentication Header, Encapsulating Security Payload, and Internet Security Association and Key Management Protocol.
- AH provides only integrity and authentication; ESP provides confidentiality, integrity, and authentication.
- IPSec may be run in tunnel mode (mainly for gateway-to-gateway connections) or in transport mode (mainly for host-to-host communications).
- The SSL protocol provides connection-oriented secure Web communications; the Secure-HTTP protocol provides connectionless secure Web communications.

6.6 Key Terms

Authentication Header (AH): An IPSec protocol designed to provide only integrity and authentication for packets transiting a network.

Encapsulating Security Payload (ESP): An IPSec protocol designed to provide confidentiality, integrity, and authentication for packets transiting a network.

encapsulation: The process used by layers of the OSI model to add layer-specific data to a packet header.

fragmentation: The process used by the Internet Protocol to divide packets into manageable fragments for transmission across varying networks.

Internet Control Message Protocol (ICMP): An administrative protocol used to transmit control messages between hosts.

Internet Protocol (IP): A Network-layer protocol used to route network traffic across the Internet and internal networks.

Internet Protocol Security (IPSec): A suite of security enhancements used to provide confidentiality, integrity, and/or authentication to the TCP/IP suite.

Internet Security Association and Key Management Protocol (ISAKMP): An IPSec protocol responsible for the creation and maintenance of security associations.

packet: A basic unit of data carried across a network by the TCP protocol.

port: An integer value that uniquely identifies a process running on a system and is used to establish interprocess communications.

Secure HTTP (HTTP-S): A secure Web communications protocol used to establish connectionless communications.

Secure Sockets Layer (SSL): A secure Web communications protocol used to establish connection-oriented communications.

socket: A combination of an IP address and a port used to uniquely identify process/system pairings.

Transmission Control Protocol (TCP): A Transport-layer protocol that runs on top of IP to provide connection-oriented communications.

transport mode: An IPSec mode that encrypts only the packet payload, facilitating host-to-host transmission over a non-IPSec network.

tunnel mode: An IPSec mode that encrypts entire packets for link-to-link communications security.

User Datagram Protocol (UDP): A Transport-layer protocol that runs on top of IP to provide connectionless communications with low overhead.

6.7 Challenge Questions

6.1 Which organization is responsible for maintaining the Request for Comments (RFC) documents that define Internet protocol standards?

a. DOD

b. IANA

c. InterNIC

d. IETF

6.2 What protocol is responsible for fragmenting datagrams that exceed the maximum length permissible on a network?

a. TCP

b. UDP

c. ICMP

d. IP

6.3 Network X receives a datagram consisting of 3,429 bytes of data. The maximum datagram length for Network X is 1,024 bytes. What will be the offset value of the last fragment?

a. 0

b. 1023

c. 1024

d. 3095

e. 3096

6.4 How many packets are used in the typical handshaking process that establishes a TCP connection?

a. 1

b. 2

c. 3

d. 4

6.5 Which two flags are used in the three-way handshaking process that establishes a TCP connection?

a. SYN

b. RST

c. FIN

d. ACK

6.6 Which one of the following terms is not normally used to describe the Transmission Control Protocol (TCP)?

a. Reliable

b. Connectionless

c. Error-checking

d. Guaranteed delivery

6.7 Which protocol is utilized by the **ping** command to determine whether a host is active on a network?

a. UDP

b. ICMP

c. FTP

d. IP

6.8 What layer of the OSI model includes the encryption and decryption of data transmitted over the network?

a. Application

b. Presentation

c. Transport

d. Data Link

6.9 What type of malicious activity involves carefully constructing a series of actions to interfere with the way memory is allocated in a system?

a. SYN flood

b. Session hijacking

c. Buffer overflow

d. Spoofing

6.10 Which layer of the OSI model may contain vulnerabilities that make a system susceptible to fragmentation attacks?

a. Session

b. Transport

c. Network

d. Data Link

6.11 If two computers are configured identically and attached to networks that differ only in the fact that one network utilizes twisted-pair cabling and the other uses fiber-optic cabling, which layer(s) of the OSI model are different between the two systems? (Choose all that apply.)

a. Network

b. Physical

c. Data Link

d. Transport

6.12 When moving a computer from one network to another, which of the following characteristics may change? (Choose all that apply.)

a. Computer name

b. IP address

c. Physical location

d. MAC address

6.13 Jim suspects that a fragmentation attack may be taking place on his network and wishes to conduct packet analysis to diagnose the

problem. What specific headers will provide him with information useful in detecting a fragmentation attack? (Choose all that apply.)

a. Fragment number

b. Total length

c. Offset

d. Type of service

6.14 Which two flags are used in the three-way handshaking process used to terminate a TCP connection?

a. SYN

b. RST

c. FIN

d. ACK

6.15 What IPSec protocol is responsible for the creation and maintenance of security associations between communicating devices?

a. ESP

b. IPcomp

c. AH

d. ISAKMP

6.16 Beth is considering implementing IPSec on her network. Her primary concern is to protect the contents of packets on the network from prying eyes. Which one of the following IPSec protocols is directly responsible for providing this type of service?

a. ESP

b. IPcomp

c. AH

d. ISAKMP

6.17 IPSec's ESP protocol may be run in two different modes. _____ mode is commonly used for host-to-host

communication across a network that may not necessarily support IPSec, whereas _____ mode is used for gateway-to-gateway communications, such as between two firewalls.

6.18 Richard suspects that a denial of service attack is taking place on his network that utilizes a large amount of SSL communications. He wishes to monitor network activity using a packet sniffer to determine whether this traffic is present on his network. What destination port should he look for in the packet sniffer output to confirm the presence of this traffic?

a. 80

b. 110

c. 443

d. 8088

6.19 Which one of the following Web security solutions is supported by most popular Web browsers?

a. SSL

b. SET

c. S-HTTP

d. SAM

6.20 Which one of the following Web security solutions is designed to function in a connectionless manner?

a. SSL

b. SET

c. S-HTTP

d. SAM

WARNING

Monitoring traffic on a network without permission probably violates the acceptable use policy for that network and may even violate one or more laws. Be sure that you have clearance from your instructor before attempting this exercise. If possible, it should be done on a closed network in a testing environment. If it is not possible to complete this exercise using packet sniffing software, your instructor may provide you with example packets to study.

6.8 Challenge Exercises

Challenge Exercise 6.1

In this exercise, you use network sniffing software to capture and analyze packets on a real computer network. You need a computer running packet sniffing software (such as Ethereal or WildPackets EtherPeek) that is attached to a network. The installation of this software normally requires

administrator access to the computer and should be completed by your instructor.

6.1 Following the instructions provided with your packet sniffing software, capture three individual packets. When capturing these packets, do not capture them consecutively; rather, capture three packets individually at random times. This will increase the likelihood that you capture a variety of packets.

6.2 View each of these packets in their binary form and analyze their content.

6.3 Describe the contents of each packet in general terms, answering the following questions:

a. What type of packet is it?

b. What is the source of the packet?

c. What is the packet's destination?

d. What type of activity likely generated this packet?

TIP

Remember to convert from binary to decimal in a manner appropriate for each type of header that you analyze. In order to complete this analysis, you should reference the appropriate RFC document(s) for each packet type that you analyze. See the "Request for Comments" sidebar earlier in this chapter for more information.

Challenge Exercise 6.2

In this exercise, you examine the contents of a digital certificate used for SSL communication with a Web site. You need a computer with a Web browser and Internet access. The following instructions describe the sequence of steps required if you are using Microsoft Internet Explorer 6.0. If you are using a different browser, you will still be able to complete this exercise, but the process may vary slightly.

6.1 Open your Web browser and access a secure Web site (i.e., one that begins with the https:// prefix). For example, go to *https://secure.safaribooksonline.com/promo.asp2.code=ITT03&portal=informit*.

6.2 If you are required to enter a username and password, please do so. Accept any security messages issued by your browser.

6.3 Once you have reached the secure Web page, pull down the File menu of your browser and select the Properties option.

6.4 Click the Certificates button in the lower-right corner of the resulting pop-up window.

6.5 Spend some time reading the information presented in the General tab and use it to answer the following questions:

a. Who issued this certificate?

b. To whom was the certificate issued?

c. What is the purpose of this certificate?

6.6 Select the Details tab of the certificate Properties window. Read the information presented and use it to answer the following questions:

a. What is the validity period of the certificate?

b. What algorithm was used to sign the certificate?

c. What algorithm was used to create the thumbprint (another term for message digest)?

d. What is the length of the key used to sign the certificate?

6.7 Close your Web browser.

6.9 Challenge Scenario

Challenge Scenario 6.1

You are the security administrator for SoftWear, a maker of T-shirts with witty computer slogans on them. SoftWear currently has four offices: a corporate headquarters in New York, a manufacturing plant in Hong Kong, and sales offices in Los Angeles and London. These offices currently communicate with each other using unsecured IP communication over the Internet. You have been given the responsibility of adding security to these communications and have selected IPSec as the appropriate methodology to achieve that.

Currently, each office has a firewall sitting on the perimeter of its network that serves as the gateway to the Internet. These firewalls support IPSec communications. It is important to you that you provide confidentiality, integrity, and authentication across the interoffice links. Additionally, you wish to provide integrity and authentication to communications that take place between systems within an office.

Describe an IPSec implementation that meets the listed business objectives and technical requirements.