

CHAPTER

8

Security



Most Important Ideas and Concepts from Chapter 8

- ◆ **Networks are vulnerable to attacks.** Computer networks, and in particular the Internet, are vulnerable to a wide array of attacks. These attacks include the following:
 - **Sniffing:** Also known as eavesdropping and wire tapping, the act of sniffing is to make copies of packets as they travel across a link. The packets can then be examined for sensitive information. It is trivial to sniff packets sent over a wireless LAN; sniffing packets sent between Alice and Bob over a wired link can be more challenging. If the nodes in an Ethernet LAN are interconnected with a hub, then sniffing is straightforward; however, if they are interconnected with a switch, then sniffing is difficult (but not impossible). The primary defense against sniffing is encryption.
 - **Modification, insertion, and deletion of message content:** Even more insidious than sniffing, here the attacker modifies the transmitted data without either of the communicating entities being aware of it. This can be done by flipping bits, inserting bits or packets, or deleting bits or packets. As described in the chapter, the primary defenses against data modification are message digests, sequence numbers, and encryption.
 - **Masquerading:** The attacker pretends to be something that it is not (a person, a router interface, and so on). For example, Trudy may pretend to be Alice, who normally is trusted by Bob. Trudy can then send commands to Bob—who thinking that the commands come from trustworthy Alice—executes the commands. Or Trudy, masquerading as Alice, may be able to extract sensitive information from Bob. As described in the chapter, defenses for masquerading include handshaking, certificates, nonces, and message digests.
 - **Network mapping:** Before attacking a particular network, such as a corporate network, often attackers would like to know the IP addresses of machines on the network, the operating systems they use, and the services that they offer. With this information, attacks can be more focused and are less likely to cause alarm. The process of gathering this information is known as network mapping. A ping sweep can be used to determine the IP addresses of the operational hosts on the network by simply observing which addresses respond to a ping message. Port scanning refers to the technique of sequentially contacting (either via a TCP connection request, or via a simple UDP datagram) port numbers on a machine and seeing what happens in response. These responses, in turn, can be used to determine the services offered (for example, HTTP or FTP) by the machine. Attackers also use traceroute to attempt to map the target network. The primary defenses against network mapping are firewalls and intrusion detection systems.
 - **Denial of service attacks:** Typically, a DoS attack works by creating so much work for the infrastructure under attack that legitimate work cannot be performed. In a SYN flood attack, the attacker deluges a server with TCP SYN

packets, each having a spoofed IP source address. The server, not being able to differentiate between a legitimate SYN and a spoofed SYN, completes the second step of the TCP handshake for a spoofed SYN, allocating data structures and state. The third step of the three-way handshake is never completed by the attacker, leaving an ever-increasing number of partially open connections, and eventually bringing the system to its knees.

- **Distributed denial of service attack (DDoS):** The attacker first gains access to numerous hosts across the Internet (for example, via self-propagating worms) and inserts slave programs in these hosts. Once a large number of such slave programs are running, a master program contacts and instructs each of them to launch a DoS attack directed at the same target host. The resulting coordinated attack is particularly devastating, since it is coming from many directions at once. The primary defenses are over-provisioned bandwidth and upstream rate limiting.
- ◆ **Defending against attacks: packet filtering.** Packet filters, typically located at gateways to institutional networks, operate by first parsing datagram headers and then applying filtering rules from an administrator-specified rule set to determine whether to drop the datagram. Filtering decisions are typically based on (i) IP source or destination address; (ii) TCP or UDP source and destination port; (iii) ICMP message type; and (iv) connection-initialization datagrams using the TCP SYN or ACK bits. Packet filters can defend against network mapping and DDoS attacks.
- ◆ **Defending against attacks: cryptography.** There are two classes of cryptography algorithms: symmetric-key cryptography and public-key cryptography. With a symmetric cipher, a message and a key are supplied as input to an encryption algorithm, producing ciphertext. The receiver inputs the ciphertext and key into a decryption algorithm to recover the original plaintext. Importantly, the algorithm is not secret and is known to all. However, the key is known only to the communicating entities. Among the symmetric key algorithms, the textbook focuses on so-called “block ciphers,” for which DES and 3DES are examples. In a block cipher, the message is chopped into blocks (for example, 64 bits) and each block is separately encrypted. Thus, (ignoring cipher block chaining), the sender inputs a block and the key into the encryption algorithm to generate an encrypted block. The receiver inputs the encrypted block and the same key into a decryption algorithm to decrypt the block.
- ◆ **Distributing keys over a network: public-key cryptography.** In a networking environment, with communicating entities often residing in different continents, it is a non-trivial task to distribute the symmetric shared key among the communicating entities. Public-key cryptography can be used for this task. In public-key cryptography, each communicating entity independently generates two keys: a private key and a public key. Each entity makes its public-key available publicly (for example, on a Web page), but keeps its private key private, not showing it to any other entity. Bob and Alice can obtain the same shared key for symmetric-key

cryptography as follows: Alice first creates a random key, which will be the shared symmetric key. She now needs to send this key to Bob over the network in such a way that no one else can see it. To this end, she encrypts the symmetric key with Bob's public key, and sends the encrypted message to Bob. To decrypt the message and extract the symmetric key, one needs to apply to the message Bob's private key, which only Bob has. Bob decrypts the key, so that Alice and Bob finally share the same symmetric key. Distributing a symmetric key among communicating entities is a common application of public-key cryptography. There are many applications of public-key cryptography, some of which are discussed in the textbook.

- ◆ **End-point authentication.** Suppose you want to communicate with Bob, whose IP address you know. You send a message to Bob and Bob responds to you. But how do you know for sure that you are really communicating with Bob? Examining the source IP address is not sufficient, since Trudy can easily spoof Bob's IP address. Protocols for end-system authentication, using nonces and cryptography, are outlined in the textbook.
- ◆ **Message integrity.** When Bob receives a message from Alice, how does he know that the message hasn't been tampered with, that it is indeed the same message that Alice originally sent? A popular solution to this problem is for Alice to append a message digest to the message. For example, Alice may create the SHA-1 hash of the original message, encrypt it with her private key, and then append the result to the original message. Bob can decrypt the message digest with Alice's public key, and then compare the result with the hash of the received message. If the two are the same, Bob knows that the message he received is indeed the message that Alice sent.
- ◆ **Secure e-mail: PGP.** PGP is an example of securing an application-layer protocol. The sender applies symmetric-key cryptography to encrypt the message, public-key cryptography to distribute the secret key to the receiver, and a hash function for message integrity.
- ◆ **Securing a TCP connection: SSL.** Secure sockets layer (SSL) is software that sits between the application layer and TCP on both the client and server sides. Roughly speaking, when one side of the application writes data to the SSL-enhanced TCP connection, SSL encrypts the data and passes it to TCP; SSL at the other side receives data from TCP, decrypts the data, and passes the decrypted data to the other side of the application. Thus, SSL secures a TCP connection. More specifically, for data transfer, SSL creates SSL records, which include the encrypted data along with a message digest for integrity. The SSL records are sent over TCP. Before data transfer, but after the TCP connection is established, there is an SSL handshake, during which the two sides authenticate each other, agree on the cipher schemes to be employed during the SSL session, and establish session keys.
- ◆ **Secure network layer: IPsec.** IPsec is a suite of protocols that provide security at the network layer. When a chunk of data is sent between two IPsec-enabled hosts, the data is encrypted and a message digest is appended to the data (ensur-

ing data integrity). No matter what the data is—a TCP segment, a UDP segment, or an ICMP message—the data enjoys the blanket coverage provided by IPsec.

- ◆ **Securing wireless links: WEP and WPA.** The IEEE 802.11 WEP protocol provides authentication and data encryption between a host and a wireless access point (that is, base station) using a symmetric shared key approach. WEP does not specify a key management algorithm, so it is assumed that the host and wireless access point have somehow agreed on the key via an out-of-band method. Authentication is carried out as in the ap4.0 protocol that we developed in Section 8.3. Encryption is done with the RC4 stream cipher, with a different Initialization Vector (IV) used for each frame. WEP, although extensively used, has serious security flaws. A more recent protocol, WPA, is similar in many ways to WEP but is much more secure.



Review Questions

This section provides additional study questions. Answers to each question are provided in the next section.

1. **IPsec.** True or False? Consider sending a stream of packets from Host A to Host B using IPsec. Typically, a new SA will be established for each packet sent in the stream.
2. **IPsec.** True or False? Suppose that TCP is being run over IPsec. If TCP retransmits the same segment, then the encapsulating IP datagrams will have the same sequence number in the IPsec headers.
3. **SSL.** True or False? Suppose Alice and Bob are communicating over an SSL session. Suppose an attacker, who does not have any of the shared keys, inserts a bogus TCP segment into a packet stream with correct TCP checksum and sequence numbers (and correct IP addresses and port numbers). SSL at the receiving side will accept the bogus packet and pass the payload to the receiving application.
4. **Network mapping.** True or False? Suppose you are doing a traceroute from host A to host B, and all routers along the path are configured to never send ICMP messages. Then it is impossible to determine from traceroute the number of routers in the path between A and B.
5. **Port scanning.** True or False? When an attacker does a port scan to determine the open TCP ports on a target host, all of the packets that the attacker sends can have a spoofed IP address.
6. **TCP hijacking.** Suppose Alice and Bob are interacting via a TCP session, and that Trudy is on a broadcast segment where traffic passes between Alice and Bob.
 - a. In 40 words or less, describe how Trudy can masquerade as Alice and hijack the session.
 - b. In 40 words or less, why will Bob quickly drop the TCP session with the basic approach?
7. **Public-key cryptography.** Consider RSA with $p = 17$ and $q = 11$.
 - a. What are n and z ?
 - b. Let e be 7. Why is this an acceptable choice for e ?
 - c. Find d such that $de = 1 \pmod{160}$ and $d < 160$.
 - d. Encrypt the message m with $m = 88$ using the key (n, e) . Let c denote the corresponding ciphertext. Show all work. Hint: To simplify the calculations, use the following fact:

$$[(a \bmod n) \times (b \bmod n)] \bmod n = (a \times b) \bmod n$$
 - e. Decrypt c . Show all work. Again use the above fact.

8. **SSL.** Consider the Ethereal screenshot shown below for a portion of an SSL session.

No.	Time	Source	Destination	Protocol	Info
106	21.805705	128.238.38.162	216.75.194.220	SSLv2	Client Hello
108	21.830201	216.75.194.220	128.238.38.162	SSLv3	Server Hello,
111	21.853520	216.75.194.220	128.238.38.162	SSLv3	Certificate, Server Hello Done
112	21.876168	128.238.38.162	216.75.194.220	SSLv3	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
113	21.945667	216.75.194.220	128.238.38.162	SSLv3	Change Cipher Spec, Encrypted Handshake Message
114	21.954189	128.238.38.162	216.75.194.220	SSLv3	Application data
122	23.480352	216.75.194.220	128.238.38.162	SSLv3	Application data

Frame 112 (258 bytes on wire (258 bytes captured))

- Ethernet II, Src: ibm10:60:99 (00:09:6b:10:60:99), Dst: All-HSRP-routers_00 (00:00:0c:07:ac:00)
- Internet Protocol, Src: 128.238.38.162 (128.238.38.162), Dst: 216.75.194.220 (216.75.194.220)
- Transmission Control Protocol, Src Port: 2271 (2271), Dst Port: https (443), Seq: 79, Ack: 2785, Len: 204
- Secure Socket Layer
 - SSLv3 Record Layer: Handshake Protocol: Client Key Exchange
 - Content Type: Handshake (22)
 - Version: SSL 3.0 (0x0300)
 - Length: 132
 - Handshake Protocol: Client Key Exchange
 - Handshake Type: Client Key Exchange (16)
 - Length: 128
 - SSLv3 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
 - Content Type: Change Cipher Spec (20)
 - Version: SSL 3.0 (0x0300)
 - Length: 1
 - Change Cipher Spec Message
 - SSLv3 Record Layer: Handshake Protocol: Encrypted Handshake Message
 - Content Type: Handshake (22)
 - Version: SSL 3.0 (0x0300)
 - Length: 56
 - Handshake Protocol: Encrypted Handshake Message

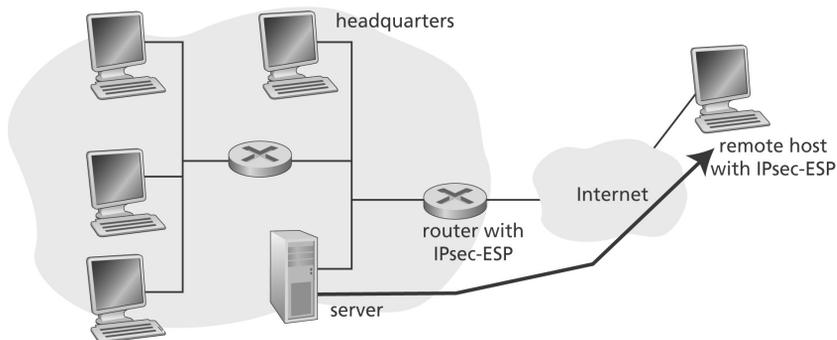
```

0030 fd 1f c2 d9 00 00 16 03 00 00 84 10 00 00 80 bd .....
0040 49 49 47 29 aa 25 90 47 7f d0 59 05 6a e7 89 56 [.....]G..V...
0050 c7 7b 12 af 08 b4 7c 60 9e 61 f1 04 b0 fb f8 3e [.....]a....>
0060 41 c0 8d c9 10 93 9c ad 1e ce 82 e0 dd e2 50 b9 A.....P.
0070 9b 4b 51 c7 3f bd ee cd 92 c4 27 5d ff dd fb 95 .KQ?...].
0080 42 3d a4 b7 71 ee c0 ff c3 ce b2 ed 60 90 6c d7 B=.q.....l.
0090 04 6e 5a 00 98 2e 52 ee b5 bc d1 c4 f5 63 f0 e3 .nZ..R.....c.
00a0 44 20 f1 c6 ba 64 58 79 46 98 3e c4 fd d7 9b 7a D)...xyF...z
00b0 02 04 09 32 f6 1d 7a a1 2d cf d2 1a 18 64 29 14 ...Z...d].
00c0 03 00 00 01 01 16 03 00 00 38 29 a9 dc 11 5a 74 .....8)...2t
00d0 7a 41 48 15 4f 50 4b e2 df 0c d0 5b c4 44 a8 e8 2AH.OPK...[.D..

```

- Is Ethernet packet 112 sent by the client or server?
- What is server's IP address and port number?
- Assuming no loss and no retransmissions, what will be the sequence number of the next TCP segment sent by the client?
- How many SSL records does Ethernet packet 112 contain?
- Does packet 112 contain a master secret or an encrypted master secret or neither?
- Assuming that the handshake type field is one byte and each length field is three bytes, what are the values of the first and last bytes of the master secret (or encrypted master secret)?
- The client encrypted handshake message takes into account how many SSL records?
- The server encrypted handshake message takes into account how many SSL records?

9. **IPsec.** Consider the network below. Suppose that the remote host and server communicate with the ESP protocol in tunnel mode.
- An IP datagram emitted by the server and destined to the remote host will have whose IP address for the source address and whose IP address for the destination address?
 - When this datagram arrives at the router, the router may or may not transform it into another IP datagram. The IP datagram sent by the router will have whose IP address for the source address and whose IP address for the destination address?



10. **PGP.** Suppose Alice sends a message m to Bob using PGP with integrity and confidentiality. Using (some or all of) the notation m , K_S , $K_S(\cdot)$, $K_B^+(\cdot)$, $K_B^-(\cdot)$, $K_A^+(\cdot)$, $K_A^-(\cdot)$, $H(\cdot)$, $H(m)$, describe what Bob does to authenticate and decrypt the message.



Answers to Review Questions

1. False. After establishing SAs to each other, all the packets sent between the hosts will use these SAs with the same session keys.
2. False. IPsec at the source increments the sequence number every time the source sends a new datagram. Since the two (identical) segments will be sent within different diagrams, the two datagrams will have different sequence numbers.
3. False. SSL sends records, each of which includes a message digest. To create the message digest, one needs a shared key, which the attacker doesn't have.
4. False. With traceroute, the source host sends a series of packets, incrementing the TTL for each packet. Eventually one of the packets reaches the destination host; when it does, the destination message sends back an error message. The source host can determine the number of intermediate routers from the TTL of the packet that reached the destination host.
5. False. To determine if a TCP port is open, the attacker sends a TCP SYN segment and waits for a response (a TCP SYN/ACK segment if the port is open). If the source IP address is spoofed in the TCP SYN segment, then the response will go to the spoofed address rather than to the attacker's IP address. Thus the attacker will never know if a SYN/ACK was sent or not, and thus will not learn if the port is open.
6.
 - a. Trudy can sniff the packets sent between Alice and Bob and determine the IP addresses and port numbers they are using. Trudy can also sniff the sequence numbers in Alice's packets and determine the sequence number for Alice's next packet. Trudy can then masquerade as Alice by sending Bob segments with the correct port numbers, sequence numbers, and acknowledgement numbers.
 - b. Bob will acknowledge Trudy's packets; Alice will see the acknowledgements for data that she never sent; Alice will send new acknowledgements, corresponding to the bytes she actually sent; Bob will receive two sets of inconsistent acknowledgments (from Alice and Trudy) and drop the connection.
7.
 - a. $n = pq = 187$; $z = (p - 1)(q - 1) = 160$
 - b. $e = 7$ is acceptable since it is relative prime with $z = 160$.
 - c. $d = 23$
 - d. $c = m^e \bmod n = 88^7 \bmod 187 = 11$
 - e. $m = c^d \bmod n = 11^{23} \bmod 187 = 88$

8.
 - a. Sent by client.
 - b. The server's IP address is 216.75.194.220 and port number is 443.
 - c. The sequence number of the next TCP segment sent by the client will be $79 + 204 = 283$.
 - d. Ethernet packet 112 contains 3 SSL records.
 - e. Packet 112 contains the encrypted secret.
 - f. First byte: BC; last byte: 29
 - g. The client encrypted handshake message takes into account 6 SSL records.
 - h. The server encrypted handshake message takes into account 8 SSL records.
9.
 - a. Source address = server's IP address; destination address = remote host's IP address.
 - b. Source address = router interface IP address; destination address = remote host's IP address.
10. Bob first extracts $K_B^+(K_S)$ and $K_S(m + K_A^-(H(m)))$ from the "package" he receives from Alice. He then applies $K_B^-()$ to $K_B^+(K_S)$ to obtain the session key K_S . He then applies K_S to $K_S(m + K_A^-(H(m)))$ to obtain both m and $K_A^-(H(m))$. He now has the unencrypted message, but it remains to verify the message's integrity. To this end, he applies $K_A^+()$ to $K_A^-(H(m))$ to obtain $x = H(m)$. He then applies $H()$ to the received message m . If the result is x , he concludes that the message is authentic.