SECTION 8.8

R29. Stateful packet filters maintain two data structures. Name them and briefly describe what they do.

R30. Consider a traditional (stateless) packet filter. This packet filter may filter packets based on TCP flag bits as well as other header fields. True or False?

R31. In a traditional packet filter, each interface can have its own access control list. True or False?

R32. Why must an application gateway work in conjunction with a router filter to be effective?

R33. Signature-based IDSs and IPSs inspect into the payloads of TCP and UDP segments. True or False?

Problems

P1. Using the monoalphabetic cipher in Figure 8.3, encode the message "This is an easy problem." Decode the message "rmij'u uamu xyj."

P2. Show that Trudy's known-plaintext attack, in which she knows the (ciphertext, plaintext) translation pairs for seven letters, reduces the number of possible substitutions to be checked in the example in Section 8.2.1 by approximately 10^9.

P3. Consider the polyalphabetic system shown in Figure 8.4. Will a chosen-plaintext attack that is able to get the plaintext encoding of the message "The quick brown fox jumps over the lazy dog." be sufficient to decode all messages? Why or why not?

P4. Consider the block cipher in Figure 8.5. Suppose that each block cipher T_i simply reverses the order of the eight input bits (so that, for example, 11110000 becomes 00001111). Further suppose that the 64-bit scrambler does not modify any bits (so that the output value of the *m*th bit is equal to the input value of the *m*th bit). (a) With $n = 3$ and the original 64-bit input equal to 10100000 repeated eight times, what is the value of the output? (b) Repeat part (a) but now change the last bit of the original 64-bit input from a 0 to a 1. (c) Repeat parts (a) and (b) but now suppose that the 64-bit scrambler inverses the order of the 64 bits.

P5. Consider the block cipher in Figure 8.5. For a given "key" Alice and Bob would need to keep eight tables, each 8 bits by 8 bits. For Alice (or Bob) to store all eight tables, how many bits of storage are necessary? How does this number compare with the number of bits required for a full-table 64-bit block cipher?

P6. Consider the 3-bit block cipher in Table 8.1. Suppose the plaintext is 100100100. (a) Initially assume that CBC is not used. What is the resulting ciphertext? (b) Suppose Trudy sniffs the ciphertext. Assuming she knows that a 3-bit block cipher without CBC is being employed (but doesn't know the specific cipher), what can she surmise? (c) Now suppose that CBC is used with IV = 111. What is the resulting ciphertext?

P7. (a) Using RSA, choose $p = 3$ and $q = 11$, and encode the word "dog" by encrypting each letter separately. Apply the decryption algorithm to the encrypted version to recover the original plaintext message. (b) Repeat part (a) but now encrypt "dog" as one message m.

P8. Consider RSA with $p = 5$ and $q = 11$.

a. What are n and z.

b. Let e be 3. Why is this an acceptable choice for e?

c. Find d such that $de = 1$ (mod z) and $d < 160$.

d. Encrypt the message $m = 8$ using the key (n, e). Let c denote the corresponding ciphertext. Show all work. *Hint:* To simplify the calculations, use the fact:

$$[(a \bmod n) \bullet (b \bmod n)] \bmod n = (a \bullet b) \bmod n$$

P9. In this problem, we explore the Diffie-Hellman (DH) public-key encryption algorithm, which allows two entities to agree on a shared key. The DH algorithm makes use of a large prime number p and another large number g less than p. Both p and g are made public (so that an attacker would know them). In DH, Alice and Bob each independently choose secret keys, S_A and S_B, respectively. Alice then computes her public key, T_A, by raising g to S_A and then taking mod p. Bob similarly computes his own public key T_B by raising g to S_B and then taking mod p. Alice and Bob then exchange their public keys over the Internet. Alice then calculates the shared secret key S by raising T_B to S_A and then taking mod p. Similarly, Bob calculates the shared key S' by raising T_A to S_B and then taking mod p.

a. Prove that in general that Alice and Bob obtain the same symmetric key, that is, prove $S = S'$.

b. With $p = 11$ and $g = 2$, suppose Alice and Bob choose private keys $S_A = 5$ and $S_B = 12$, respectively. Calculate Alice's and Bob's public keys, T_A and T_B. Show all work.

c. Following up on part (b), now calculate S as the shared symmetric key. Show all work.

d. Provide a timing diagram that shows how Diffie-Hellman can be attacked by a man-in-the-middle. The timing diagram should have three vertical lines, one for Alice, one for Bob, and one for the attacker Trudy.

P10. Suppose Alice wants to communicate with Bob using symmetric key cryptography using a session key K_S. In Section 8.2, we learned how public-key cryptography can be used to distribute the session key from Alice to Bob. In this problem, we explore how the session key can be distributed—without public key cryptography—using a key distribution center (KDC). The KDC is a server that shares a unique secret symmetric key with each registered user. For Alice and Bob, denote these keys by $K_{A\text{-}KDC}$ and $K_{B\text{-}KDC}$. Design a scheme that uses the KDC to distribute K_S to Alice and Bob. Your scheme should use three messages to distribute the session key: a message from Alice to the KDC; a message from the KDC to Alice; and finally a message from Alice to Bob. The first message is $K_{A\text{-}KDC}$ (A, B). Using the notation, $K_{A\text{-}KDC}$, $K_{B\text{-}KDC}$, S, A, *and* B answer the following questions.

a. What is the second message?

b. What is the third message?

P11. Compute a third message, different from the two messages in Figure 8.8, that has the same checksum as the messages in Figure 8.8.

P12. Suppose Alice and Bob share two secret keys: an authentication key S_1 and a symmetric encryption key S_2. Augment Figure 8.9 so that both integrity and confidentiality are provided.

P13. In the BitTorrent P2P file distribution protocol (see Chapter 2), the seed breaks the file into blocks, and the peers redistribute the blocks to each other. Without any protection, an attacker can easily wreak havoc in a torrent by masquerading as a benevolent peer and sending bogus blocks to a small subset of peers in the torrent. These unsuspecting peers then redistribute the bogus blocks to other peers, which in turn redistribute the bogus blocks to even more peers. Thus, it is critical for BitTorrent to have a mechanism that allows a peer to verify the integrity of a block, so that it doesn't redistribute bogus blocks. Assume that when a peer joins a torrent, it initially gets a `.torrent` file from a *fully* trusted source. Describe a simple scheme that allows peers to verify the integrity of blocks.

P14. The OSPF routing protocol uses a MAC rather than digital signatures to provide message integrity. Why do you think a MAC was chosen over digital signatures?

P15. Consider our authentication protocol in Figure 8.16 in which Alice authenticates herself to Bob, which we saw works well (i.e., we found no flaws in it). Now suppose that while Alice is authenticating herself to Bob, Bob must authenticate himself to Alice. Give a scenario by which Trudy, pretending to be Alice, can now authenticate herself to Bob as Alice. (*Hint*: Consider that the sequence of operations of the protocol, one with Trudy initiating and one with Bob initiating, can be arbitrarily interleaved. Pay particular attention to the fact that both Bob and Alice will use a nonce, and that if care is not taken, the same nonce can be used maliciously.)

P16. In the man-in-the-middle attack in Figure 8.19, Alice has not authenticated Bob. If Alice were to require Bob to authenticate himself using the public-key authentication protocol, would the man-in-the-middle attack be avoided? Explain your reasoning.

P17. Figure 8.20 shows the operations that Alice must perform with PGP to provide confidentiality, authentication, and integrity. Diagram the corresponding operations that Bob must perform on the package received from Alice.

P18. Suppose Alice wants to send an e-mail to Bob. Bob has a public-private key pair (K_B^+, K_B^-), and Alice has Bob's certificate. But Alice does not have a public, private key pair. Alice and Bob (and the entire world) share the same hash function $H(\cdot)$.

a. In this situation, is it possible to design a scheme so that Bob can verify that Alice created the message? If so, show how with a block diagram for Alice and Bob.

b. Is it possible to design a scheme that provides confidentiality for sending the message from Alice to Bob? If so, show how with a block diagram for Alice and Bob.

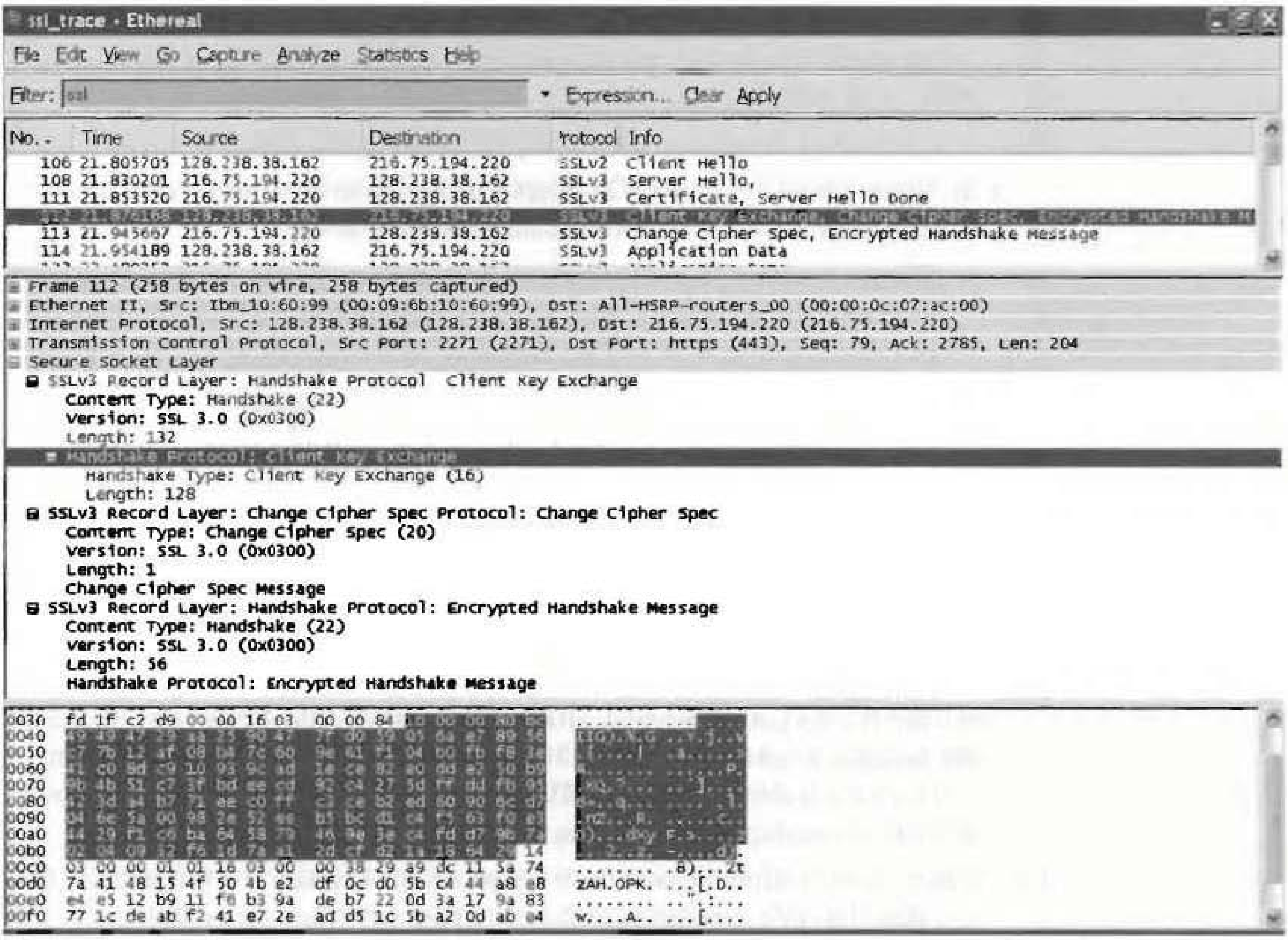
P19. Consider the Wireshark output below for a portion of an SSL session.

a. Is Wireshark packet 112 sent by the client or server?

b. What is the server's IP address and port number?

c. Assuming no loss and no retransmissions, what will be the sequence number of the next TCP segment sent by the client?

d. How many SSL records does Wireshark packet 112 contain?



e. Does packet 112 contain a Master Secret or an Encrypted Master Secret or neither?

f. Assuming that the handshake type field is 1 byte and each length field is 3 bytes, what are the values of the first and last bytes of the Master Secret (or Encrypted Master Secret)?

g. The client encrypted handshake message takes into account how many SSL records?

h. The server encrypted handshake message takes into account how many SSL records?

P20. In Section 8.5.1, it is shown that without sequence numbers, Trudy (a woman-in-the middle) can wreak havoc in an SSL session by interchanging TCP segments. Can Trudy do something similar by deleting a TCP segment? What does she need to do to succeed at the deletion attack? What effect will it have?

P21. 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). Will SSL at the receiving side accept the bogus packet and pass the payload to the receiving application? Why or why not?

P22. The following True/False questions pertain to Figure 8.29.

a. When a host in 172.16.1/24 sends a datagram to an Amazon.com server, the router R1 will encrypt the datagram using IPsec.

b. When a host in 172.16.1/24 sends a datagram to a host in 172.16.2/24, the router R1 will change the source and destination address of the IP datagram.

c. Suppose a host in 172.16.1/24 initiates a TCP connection to a Web server in 172.16.2/24. As part of this connection, all datagrams sent by R1 will have protocol number 50 in the left-most IPv4 header field.

d. Consider sending a TCP segment from a host in 172.16.1/24 to a host in 172.16.2/24. Suppose the acknowledgment for this segment gets lost, so that TCP resends the segment. Because IPsec uses sequence numbers, R1 will not resend the TCP segment.

P23. Consider the example in Figure 8.29. Suppose Trudy is a woman-in-the-middle, who can insert datagrams into the stream of datagrams going from R1 and R2. As part of a replay attack, Trudy sends a duplicate copy of one of the datagrams sent from R1 to R2. Will R2 decrypt the duplicate datagram and forward it into the branch-office network? If not, describe in detail how R2 detects the duplicate datagram.

P24. Consider the following pseudo-WEP protocol. The key is 4 bits and the IV is 2 bits. The IV is appended to the end of the key when generating the keystream. Suppose that the shared secret key is 1010. The keystreams for the four possible inputs are as follows:

101000: 0010101101010101001011010100100 . . .

101001: 1010011011001010110100100101101 . . .

101010: 0001101000111100010100101001111 . . .

101011: 1111101010000000101010100010111 . . .

Suppose all messages are 8-bits long. Suppose the ICV (integrity check) is 4-bits long, and is calculated by XOR-ing the first 4 bits of data with the last 4 bits of data. Suppose the pseudo-WEP packet consists of three fields: first the IV field, then the message field, and last the ICV field, with some of these fields encrypted.

a. We want to send the message $m = 10100000$ using the IV = 11 and using WEP. What will be the values in the three WEP fields?

b. Show that when the receiver decrypts the WEP packet, it recovers the message and the ICV.

c. Suppose Trudy intercepts a WEP packet (not necessarily with the IV = 11) and wants to modify it before forwarding it to the receiver. Suppose Trudy flips the first ICV bit. Assuming that Trudy does not know the keystreams for any of the IVs, what other bit(s) must Trudy also flip so that the received packet passes the ICV check?

d. Justify your answer by modifying the bits in the WEP packet in part (a), decrypting the resulting packet, and verifying that the integrity check.

P25. Provide a filter table and a connection table for a stateful firewall that is restrictive as possible but accomplishes the following:

a. Allows all internal users to establish Telnet sessions with external hosts.

b. Allows external users to surf the company Web site at 222.22.0.12.

c. But otherwise blocks all inbound and outbound traffic.

The internal network is 222.22/16. In your solution, suppose that the connection table is currently caching three connections, all from inside to outside. You'll need to invent appropriate IP addresses and port numbers.

P26. Suppose Alice wants to visit the Web site activist.com using TOR-like service. This service uses two non-colluding proxy servers Proxy1 and Proxy2. Alice first obtains the certificates (each containing a public key) for Proxy1 and Proxy2 from some central server. Denote $K_1^+(\ )$, $K_2^+(\ )$, $K_1^-(\ )$, and $K_2^-(\ )$ for the encryption/decryption with public and private RSA keys.

a. Using a timing diagram, provide a protocol (as simple as possible) that enables Alice to establish a shared session key S_1 with Proxy1. Denote $S_1(m)$ for encryption/decryption of data m with the shared key S_1.

b. Using a timing diagram, provide a protocol (as simple as possible) that allows Alice to establish a shared session key S_2 with Proxy2 *without revealing her IP address to Proxy2.*

c. Assume now that shared keys S_1 and S_2 are now established. Using a timing diagram, provide a protocol (as simple as possible and *not using public-key cryptography*) that allows Alice to request an html page from activist.com *without revealing her IP address to Proxy2* and *without revealing to Proxy1 which site she is visiting*. Your diagram should end with an HTTP request arriving at activist.com.

Discussion Questions

D1. Suppose that an intruder could insert DNS messages into and remove DNS messages from DNS servers. Give three scenarios showing the problems that such an intruder could cause.

D2. What is Kerberos? How does it work? How does it relate with Problem P10 of this chapter?

D3. If IPsec provides security at the network layer, why is it that security mechanisms are still needed at layers above IP?

D4. Research botnets. What protocols and systems do attackers use to control and update botnets today?

Wireshark Lab

In this lab (available from the companion Web site), we investigate the Secure Sockets Layer (SSL) protocol. Recall from Section 8.5 that SSL is used for securing a TCP connection, and that it is extensively used in practice for secure Internet transactions. In this lab, we will focus on the SSL records sent over the TCP connection. We will attempt to delineate and classify each of the records, with a goal of understanding the why and how for each record. We investigate the various SSL record types as well as the fields in the SSL messages. We do so by analyzing a trace of the SSL records sent between your host and an e-commerce server.

IPsec Lab

In this lab (available from the companion Web site), we will explore how to create IPsec SAs between linux boxes. You can do the first part of the lab with two ordinary linux boxes, each with one Ethernet adapter. But for the second part of the lab, you will need four linux boxes, two of which having two Ethernet adapters. In the second half of the lab, you will create IPsec SAs using the ESP protocol in the tunnel mode. You will do this by first manually creating the SAs, and then by having IKE create the SAs.