A discussion of the problem of recognizing and defending against denial-of-service attacks can be found in Moore et al. [MVS01], Spatscheck and Peterson [SP99], and Qiexh et al. [QPP02]. Recent techniques used to identify the source of attacks can be found in papers by Savage et al. [SWKA00] and Snoeren et al. [SPS$^+$01]. The increasing threat of DDoS attacks is discussed by Garber [Gar00] and Harrison [Har00], and early approaches to defending against such attacks are reported in a paper by Park and Lee [PL01]. A novel approach to DoS prevention which falls in the "clean slate" category is the TVA approach by Yang et al. [YWA08].

Finally, we recommend the following live references:

- http://www.cert.org/: The website of CERT, an organization focused on computer security issues.
- http://www.cacr.math.uwaterloo.ca/hac/: Downloadable copy of [MvOV96] a comprehensive cryptography reference.

EXERCISES

1. Find or install an encryption utility (e.g., the Unix des command or pgp) on your system. Read its documentation and experiment with it. Measure how fast it is able to encrypt and decrypt data. Are these two rates the same? Try to compare these timing results using different key sizes; for example, compare AES with triple-DES.

2. Diagram cipher block chaining as described in Section 8.1.1.

3. Learn about a key escrow, or key surrender, scheme (for example, Clipper). What are the pros and cons of key escrow?

4. A good cryptographic hashing algorithm should produce random outputs; that is, the probability of any given hash value should be approximately the same as any other for randomly chosen input data. What would be the consequence of using a hash algorithm whose outputs were not random? Consider, for example, the case where some hash values are twice as likely to occur as others.

5. Suppose Alice uses the Needham–Schroeder authentication protocol described in Section 8.3.3 to initiate a session with Bob. Further suppose that an adversary is able to eavesdrop on the

authentication messages and, long after the session has completed, discover the (unencrypted) session key. How could the adversary deceive Bob into authenticating the adversary as Alice?

6. One mechanism for resisting replay attacks in password authentication is to use *one-time passwords*: A list of passwords is prepared, and once $password[N]$ has been accepted the server decrements N and prompts for $password[N-1]$ next time. At $N=0$ a new list is needed. Outline a mechanism by which the user and server need only remember one master password mp and have available locally a way to compute $password[N] = f(mp, N)$. Hint: Let g be an appropriate one-way function (e.g., MD5) and let $password[N] = g^N(mp) = g$ applied N times to mp. Explain why knowing $password[N]$ doesn't help reveal $password[N-1]$.

7. Suppose a user employs one-time passwords as above (or, for that matter, reusable passwords), but that the password is transmitted sufficiently slowly.

(a) Show that an eavesdropper can gain access to the remote server with a relatively modest number of guesses. (Hint: The eavesdropper starts guessing after the original user has typed all but one character of the password.)

(b) To what other attacks might a user of one-time passwords be subject?

8. The Diffie–Hellman key exchange protocol is vulnerable to a "man-in-the-middle" attack as shown in Section 8.3.4 and Figure 8.12. Outline how Diffie–Hellman can be extended to protect against this possibility.

9. Suppose we have a very short secret s (e.g., a single bit or even a Social Security number), and we wish to send someone else a message m now that will not reveal s but that can be used later to verify that we did know s. Explain why $m = \mathrm{MD5}(s)$ or $m = \mathrm{E}(s)$ with RSA encryption would not be secure choices, and suggest a better choice.

10. Suppose two people want to play poker over the network. To deal the cards they need a mechanism for fairly choosing a random number x between them; each party stands to lose if the other party can unfairly influence the choice of x. Describe such a mechanism. Hint: You may assume that if either of two bit strings x_1 and x_2 are random, then the exclusive-OR $x = x_1 \oplus x_2$ is random.

11. Estimate the probabilities of finding two messages with the same MD5 checksum, given total numbers of messages of 2^{63}, 2^{64}, and 2^{65}. Hint: This is the Birthday Problem again, as in Exercise 48 in Chapter 2, and again the probability that the $k+1$th message has a different checksum from each of the preceding k is $1 - k/2^{128}$. However, the approximation in the hint there for simplifying the product fails rather badly now. So, instead, take the log of each side and use the approximation $\log(1 - k/2^{128}) \approx -k/2^{128}$.

12. Suppose we wanted to encrypt a Telnet session with, say, 3DES. Telnet sends lots of 1-byte messages, while 3DES encrypts in blocks of 8 bytes at a time. Explain how 3DES might be used securely in this setting.

13. Consider the following simple UDP protocol (based loosely on TFTP, *Request for Comments* 1350) for downloading files:

- Client sends a file request.
- Server replies with first data packet.
- Client sends ACK, and the two proceed using stop-and-wait.

Suppose client and server possess keys K_C and K_S, respectively, and that these keys are known to each other.

(a) Extend the file downloading protocol, using these keys and MD5, to provide sender authentication and message integrity. Your protocol should also be resistant to replay attacks.

(b) How does the extra information in your revised protocol protect against the arrival of late packets from prior connection incarnations and sequence number wraparound?

14. Using the browser of your choice, find out what certification authorities for HTTPS your browser is configured by default to trust. Do you trust these agencies? Find out what happens when you disable trust of some or all of these certification authorities.

15. Use an OpenPGP implementation such as GnuPG to do the following. Note that no email is involved—you are working exclusively with files on a single machine.
    (a) Generate a public–private key pair.
    (b) Use your public key to encrypt a file, as if for secure storage, and then use your private key to decrypt it.
    (c) Use your key pair to digitally sign an unencrypted file and then, as if you were someone else, verify your signature using your public key.
    (d) Consider the first public–private key pair as belonging to Alice, and generate a second public–private key pair, for Bob. Playing the role of Alice, encrypt and sign a file intended for Bob. (Be sure to sign as Alice, not Bob.) Then, playing the role of Bob, verify Alice's signature and decrypt the file.

16. Consider a certification hierarchy as described in Section 8.2.1. A root CA signs a certificate for a second-tier CA, and the second-tier CA signs a certificate for Alice. Bob has the public key for the root CA, so he can verify the certificate of the second-tier CA. Why might Bob still not trust that the certificate for Alice truly establishes Alice as the owner of the public key in the certificate?

17. PuTTY (pronounced "putty") is a popular free SSH client—an application that implements the client side of SSH connections—for Unix and Windows. Its documentation is accessible on the Web.
    (a) How does PuTTY handle authentication of a server that it has not previously connected to?
    (b) How are clients authenticated to servers?
    (c) PuTTY supports several ciphers. How does it determine which one to use for a particular connection?
    (d) PuTTY supports ciphers, such as DES, that might be considered too weak for some—or any—situations. Why? How does PuTTY determine which ciphers are weak, and how does it use that information?
    (e) For a given connection, PuTTY lets a user specify a maximum amount of time and/or transmitted data after which PuTTY will initiate the establishment of a new session key, which the

documentation refers to as a *key exchange* or *rekeying*. What is the motivation behind this feature?

(f) Use PuTTYgen, the PuTTY key generator, to generate a public–private key pair for one of the PuTTY-supported public key ciphers.

18. Suppose you want your firewall to block all incoming Telnet connections but to allow outbound Telnet connections. One approach would be to block all inbound packets to the designated Telnet port (23).

(a) We might want to block inbound packets to other ports as well, but what inbound TCP connections *must* be permitted in order not to interfere with outbound Telnet?

(b) Now suppose your firewall is allowed to use the TCP header Flags bits in addition to the port numbers. Explain how you can achieve the desired Telnet effect here while at the same time allowing no inbound TCP connections.

19. Suppose a firewall is configured to allow outbound TCP connections but inbound connections only to specified ports. The FTP protocol now presents a problem: When an inside client contacts an outside server, the outbound TCP control connection can be opened normally but the TCP data connection traditionally is inbound.

(a) Look up the FTP protocol in, for example, *Request for Comments* 959. Find out how the PORT command works. Discuss how the client might be written so as to limit the number of ports to which the firewall must grant inbound access. Can the number of such ports be limited to one?

(b) Find out how the FTP PASV command can be used to solve this firewall problem.

20. Suppose filtering routers are arranged as in Figure 8.21; the primary firewall is R1. Explain how to configure R1 and R2 so that outsiders can Telnet to net 2 but not to hosts on net 1. To avoid “leapfrogging” break-ins to net 1, also disallow Telnet connections from net 2 to net 1.

21. Why might an Internet Service Provider want to block certain *outbound* traffic?

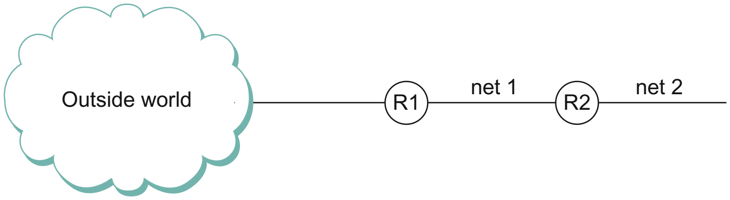

FIGURE 8.21 Diagram for Exercise 18.

22. It is said that IPsec may not work with Network Address Translation (NAT) (RFC 1631). However, whether IPsec will work with NAT depends on which mode of IPsec and NAT we use. Suppose we use true NAT, where only IP addresses are translated (without port translation). Will IPsec and NAT work in each of the following cases? Explain why or why not.

(a) IPsec uses ESP transport mode.

(b) IPsec uses ESP tunnel mode.

(c) What if we use PAT (Port Address Translation), also known as Network Address/Port Translation (NAPT) in NAT, where in addition to IP addresses port numbers will be translated to share one IP address from outside the private network?