password files, containing nearly 14,000 encrypted passwords. The result, which the author rightly characterizes as frightening, is shown in Table 11.3. In all, nearly one-fourth of the passwords were guessed. The following strategy was used: 1. Try the user's name, initials, account name, and other relevant personal information. In all, 130 different permutations for each user were tried. 2. Try words from various dictionaries. The author compiled a dictionary of over 60,000 words, including the online dictionary on the system itself, and various other lists as shown. 3. Try various permutations on the words from step 2. This included making the first letter uppercase or a control character, making the entire word uppercase, reversing the word, changing the letter "o" to the digit "zero," and so on. These permutations added another 1 million words to the list. 4. Try various capitalization permutations on the words from step 2 that were not considered in step 3. This added almost 2 million additional words to the list. Thus, the test involved in the neighborhood of 3 million words. Using the fastest Thinking Machines implementation listed earlier, the time to encrypt all these words for all possible salt values is under an hour. Keep in mind that such a thorough search could produce a success rate of about 25%, whereas even a single hit may be enough to gain a wide range of privileges on a system. Access Control One way to thwart a password attack is to deny the opponent access to the password file. If the encrypted password portion of the file is accessible only by a privileged user, then the opponent cannot read it without already knowing the password of a privileged user. [SPAF92a] points out several flaws in this strategy: ■■ Many systems, including most UNIX systems, are susceptible to unanticipated break-ins. Once an attacker has gained access by some means, he or she may wish to obtain a collection of passwords in order to use different accounts for different logon sessions to decrease the risk of detection. Or a user with an account may desire another user's account to access privileged data or to sabotage the system. ■■ An accident of protection might render the password file readable, thus compromising all the accounts. 402 chapter 11 / Intruders Type of Password Search Size Number of Matches Percentage of Passwords Matched Cost/Benefit Ratioa User/account name 130 368 2.7% 2.830 Character sequences 866 22 0.2% 0.025 Numbers 427 9 0.1% 0.021 Chinese 392 56 0.4% 0.143 Place names 628 82 0.6% 0.131 Common names 2239 548 4.0% 0.245 Female names 4280 161 1.2% 0.038 Male names 2866 140 1.0% 0.049 Uncommon names 4955 130 0.9% 0.026 Myths & legends 1246 66 0.5% 0.053 Shakespearean 473 11 0.1% 0.023 Sports terms 238 32 0.2% 0.134 Science fiction 691 59 0.4% 0.085 Movies and actors 99 12 0.1% 0.121 Cartoons 92 9 0.1% 0.098 Famous people 290 55 0.4% 0.190 Phrases and patterns 933 253 1.8% 0.271 Surnames 33 9 0.1% 0.273 Biology 58 1 0.0% 0.017 System dictionary 19683 1027 7.4% 0.052 Machine names 9018 132 1.0% 0.015 Mnemonics 14 2 0.0% 0.143 King James bible 7525 83 0.6% 0.011 Miscellaneous words 3212 54 0.4% 0.017 Yiddish words 56 0 0.0% 0.000 Asteroids 2407 19 0.1% 0.007 Total 62727 3340 24.2% 0.053 a Computed as the number of matches divided by the search size. The more words that needed to be tested for a match, the lower the cost/benefit ratio. Table 11.3 Passwords Cracked from a Sample Set of 13,797 Accounts [KLEI90] ■■ Some of the users have accounts on other machines in other protection domains, and they use the same password. Thus, if the passwords could be read by anyone on one machine, a machine in another location might be compromised. Thus, a more effective strategy would be to force users to select passwords that are difficult to guess. 11.3 / Password Management 403 Password Selection Strategies The lesson from the two experiments just described ([SPAF92a], [KLEI90]) is that, left to their own devices, many users choose a password that is too short or too easy to guess. At the other extreme, if users are assigned passwords consisting of eight randomly selected

printable characters, password cracking is effectively impossible. But it would be almost as impossible for most users to remember their passwords. Fortunately, even if we limit the password universe to strings of characters that are reasonably memorable, the size of the universe is still too large to permit practical cracking. Our goal, then, is to eliminate guessable passwords while allowing the user to select a password that is memorable. Four basic techniques are in use: ■■ User education ■■ Computer-generated passwords ■■ Reactive password checking ■■ Proactive password checking Users can be told the importance of using hard-to-guess passwords and can be provided with guidelines for selecting strong passwords. This user education strategy is unlikely to succeed at most installations, particularly where there is a large user population or a lot of turnover. Many users will simply ignore the guidelines. Others may not be good judges of what is a strong password. For example, many users (mistakenly) believe that reversing a word or capitalizing the last letter makes a password unguessable. Computer-generated passwords also have problems. If the passwords are quite random in nature, users will not be able to remember them. Even if the password is pronounceable, the user may have difficulty remembering it and so be tempted to write it down. In general, computer-generated password schemes have a history of poor acceptance by users. FIPS PUB 181 defines one of the best-designed automated password generators. The standard includes not only a description of the approach but also a complete listing of the C source code of the algorithm. The algorithm generates words by forming pronounceable syllables and concatenating them to form a word. A random number generator produces a random stream of characters used to construct the syllables and words. A reactive password checking strategy is one in which the system periodically runs its own password cracker to find guessable passwords. The system cancels any passwords that are guessed and notifies the user. This tactic has a number of drawbacks. First, it is resource intensive if the job is done right. Because a determined opponent who is able to steal a password file can devote full CPU time to the task for hours or even days, an effective reactive password checker is at a distinct disadvantage. Furthermore, any existing passwords remain vulnerable until the reactive password checker finds them. The most promising approach to improved password security is a proactive password checker. In this scheme, a user is allowed to select his or her own password. However, at the time of selection, the system checks to see if the password is allowable and, if not, rejects it. Such checkers are based on the philosophy that, with sufficient guidance from the system, users can select memorable passwords from a fairly large password space that are not likely to be guessed in a dictionary attack. 404 chapter 11 / Intruders The trick with a proactive password checker is to strike a balance between user acceptability and strength. If the system rejects too many passwords, users will complain that it is too hard to select a password. If the system uses some simple algorithm to define what is acceptable, this provides guidance to password crackers to refine their guessing technique. In the remainder of this subsection, we look at possible approaches to proactive password checking. The first approach is a simple system for rule enforcement. For example, the following rules could be enforced: ■■ All passwords must be at least eight characters long. ■■ In the first eight characters, the passwords must include at least one each of uppercase, lowercase, numeric digits, and punctuation marks. These rules could be coupled with advice to the user. Although this approach is superior to simply educating users, it may not be sufficient to thwart password crackers. This scheme alerts crackers as to which passwords not to try but may still make it possible to do password cracking. Another possible procedure is simply to compile a large dictionary of possible “bad” passwords. When a user selects a

password, the system checks to make sure that it is not on the disapproved list. There are two problems with this approach: ■■ Space: The dictionary must be very large to be effective. For example, the dictionary used in the Purdue study [SPAF92a] occupies more than 30 megabytes of storage. ■■ Time: The time required to search a large dictionary may itself be large. In addition, to check for likely permutations of dictionary words, either those words most be included in the dictionary, making it truly huge, or each search must also involve considerable processing. Bloom Filter A technique [SPAF92a, SPAF92b] for developing an effective and efficient proactive password checker that is based on rejecting words on a list has been implemented on a number of systems, including Linux. It is based on the use of a Bloom filter [BLOO70]. To begin, we explain the operation of the Bloom filter. A Bloom filter of order k consists of a set of k independent hash functions H1(x), H2(x), c , Hk(x), where each function maps a password into a hash value in the range 0 to N - 1. That is, Hi (Xj) = y 1 ... i ... k; 1 ... j ... D; 0 ... y ... N - 1 where Xj = jth word in password dictionary D = number of words in password dictionary The following procedure is then applied to the dictionary: 1. A hash table of N bits is defined, with all bits initially set to 0. 11.3 / Password Management 405 2. For each password, its k hash values are calculated, and the corresponding bits in the hash table are set to 1. Thus, if Hi (Xj) = 67 for some (i, j), then the sixty-seventh bit of the hash table is set to 1; if the bit already has the value 1, it remains at 1. When a new password is presented to the checker, its k hash values are calculated. If all the corresponding bits of the hash table are equal to 1, then the password is rejected. All passwords in the dictionary will be rejected. But there will also be some “false positives” (i.e., passwords that are not in the dictionary but that produce a match in the hash table). To see this, consider a scheme with two hash functions. Suppose that the passwords undertaker and hulkhogan are in the dictionary, but xG%#jj98 is not. Further suppose that H1(undertaker) = 25 H1(hulkhogan) = 83 H1(xG%#jj98) = 665 H2(undertaker) = 998 H2(hulkhogan) = 665 H2(xG%#jj98) = 998 If the password xG%#jj98 is presented to the system, it will be rejected even though it is not in the dictionary. If there are too many such false positives, it will be difficult for users to select passwords. Therefore, we would like to design the hash scheme to minimize false positives. It can be shown that the probability of a false positive can be approximated by: P ≈ (1 - ekD/N) k = (1 - ek/R) k or, equivalently, R ≈ -k ln(1 - P1/k) where k = number of hash functions N = number of bits in hash table D = number of words in dictionary R = N/D, ratio of hash table size (bits) to dictionary size (words) Figure 11.7 plots P as a function of R for various values of k. Suppose we have a dictionary of 1 million words and we wish to have a 0.01 probability of rejecting a password not in the dictionary. If we choose six hash functions, the required ratio is R = 9.6. Therefore, we need a hash table of 9.6 * 106 bits or about 1.2 MBytes of storage. In contrast, storage of the entire dictionary would require on the order of 8 MBytes. Thus, we achieve a compression of almost a factor of 7. Furthermore, password checking involves the straightforward calculation of six hash functions and is independent of the size of the dictionary, whereas with the use of the full dictionary, there is substantial searching.1 1 The Bloom filter involves the use of probabilistic techniques. There is a small probability that some passwords not in the dictionary will be rejected. It is often the case in designing algorithms that the use of probabilistic techniques results in a less time-consuming or less complex solution, or both. 406 chapter 11 / Intruders Review Questions 11.1 List and briefly define three classes of intruders. 11.2 Give examples of intrusion. 11.3 List the direct approaches that can be implemented to counter insider attacks. 11.4 Explain how statistical anomaly detection and rule-based intrusion detection are used to detect different types of intruders. 11.5 List

the tests that can be performed to determine if a user's current activity is statistically anomalous or whether it is within acceptable parameters. 11.6 What is the base- rate fallacy? Figure 11.7 Performance of Bloom Filter 0.001 0.01 0.1 1 Pr [false positive] 0 5 10 15 20 Ratio of hash table size (bits) to dictionary size (words) 4 hash functions 2 hash functions 6 hash functions 11.4 Key Terms, Review Questions, and Problems Key Terms audit record base-rate fallacy bloom filter distributed intrusion detection honeypot intruder intrusion detection intrusion detection exchange format password rainbow table rule-based intrusion detection salt value signature detection statistical anomaly detection 11.4 / Key Terms, Review Questions, and Problems 407 Conservativeness of signatures Frequency of alerts More speci c or stricter Less speci c or looser 11.7 List the possible locations where a honeypot can be deployed. 11.8 Briefly explain the purposes that a salt serves in the context of UNIX password management. 11.9 Discuss the threats to the UNIX password scheme. Problems 11.1 In the context of an IDS, we define a false positive to be an alarm generated by an IDS in which the IDS alerts to a condition that is actually benign. A false negative occurs when an IDS fails to generate an alarm when an alert-worthy condition is in effect. Using the following diagram, depict two curves that roughly indicate false positives and false negatives, respectively. 11.2 The overlapping area of the two probability density functions of Figure 11.1 represents the region in which there is the potential for false positives and false negatives. Further, Figure 11.1 is an idealized and not necessarily representative depiction of the relative shapes of the two density functions. Suppose there is 1 actual intrusion for every 1000 authorized users, and the overlapping area covers 1% of the authorized users and 50% of the intruders. a. Sketch such a set of density functions and argue that this is not an unreasonable depiction. b. Observe, that the overlap region equally covers authorized users and intruders. Does it always mean there is equal probability that events in this region are by authorized users and intruders? Justify your answer. 11.3 An example of a host-based intrusion detection tool is the tripwire program. This is a file integrity checking tool that scans files and directories on the system on a regular basis and notifies the administrator of any changes. It uses a protected database of cryptographic checksums for each file checked and compares this value with that recomputed on each file as it is scanned. It must be configured with a list of files and directories to check, and what changes, if any, are permissible to each. It can allow, for example, log files to have new entries appended, but not for existing entries to be changed. What are the advantages and disadvantages of using such a tool? Consider the problem of determining which files should only change rarely, which files may change more often and how, and which change frequently and hence cannot be checked. Hence consider the amount of work in both the configuration of the program and on the system administrator monitoring the responses generated. 11.4 A taxicab was involved in a fatal hit-and-run accident at night. Two cab companies, the Yellow and the Red, operate in the city. You are told that: ■ 85% of the cabs in the city are Yellow and 15% are Red. ■ A witness identified the cab as Red. 408 chapter 11 / Intruders The court tested the reliability of the witness under the same circumstances that existed on the night of the accident and concluded that the witness was correct in identifying the color of the cab 90% of the time. What is the probability that the cab involved in the incident was Red rather than Yellow? 11.5 Explain the suitability or unsuitability of the following passwords: a. anu 1998 b. 5mimf2a3c (for 5 members in my family 2 adults 3 children) c. Coimbatore16 d. Windows e. Olympics f. msk@123 g. g.0987654 h. iamking 11.6 An early attempt to force users to use less predictable passwords involved computersupplied passwords. The passwords were eight characters long and were taken from the character set consisting of lowercase

letters and digits. They were generated by a pseudorandom number generator with 215 possible starting values. Using the technology of the time, the time required to search through all character strings of length 8 from a 36-character alphabet was 112 years. Unfortunately, this is not a true reflection of the actual security of the system. Explain the problem. 11.7 Assume that passwords are selected from five-character combinations of 26 alphabetic characters. Assume that an adversary is able to attempt passwords at a rate of one per second. a. Assuming no feedback to the adversary until each attempt has been completed, what is the expected time to discover the correct password? b. Assuming feedback to the adversary flagging an error as each incorrect character is entered, what is the expected time to discover the correct password? 11.8 Assume that source elements of length k are mapped in some uniform fashion into a target elements of length p. If each digit can take on one of r values, then the number of source elements is r k and the number of target elements is the smaller number r p . A particular source element xi is mapped to a particular target element yj . a. What is the probability that the correct source element can be selected by an adversary on one try? b. What is the probability that a different source element xk(xi ≠ xk) that results in the same target element, yj , could be produced by an adversary? c. What is the probability that the correct target element can be produced by an adversary on one try? 11.9 A phonetic password generator picks two segments randomly for each six-letter password. The form of each segment is C9VC (consonant, digit, vowel, consonant), where V = 6a, e, i, o, u7 and C ≠ V. a. What is the total password population? b. What is the probability of an adversary guessing a password correctly? 11.10 Assume that passwords are limited to the use of the 95 printable ASCII characters and that all passwords are 12 characters in length. Assume a password cracker with an encryption rate of 6.4 million encryptions per second. How long will it take to test exhaustively all possible passwords on a UNIX system? 11.11 Because of the known risks of the UNIX password system, the SunOS-4.0 documentation recommends that the password file be removed and replaced with a publicly readable file called /etc/publickey. An entry in the file for user A consists of a user's identifier IDA, the user's public key, PUa, and the corresponding private key PRa. This private key is encrypted using DES with a key derived from the user's login password Pa. When A logs in, the system decrypts E(Pa, PRa) to obtain PRa. a. The system then verifies that Pa was correctly supplied. How? b. How can an opponent attack this system? 11.12 The encryption scheme used for UNIX passwords is one way; it is not possible to reverse it. Therefore, would it be accurate to say that this is, in fact, a hash code rather than an encryption of the password? 11.4 / Key Terms, Review Questions, and Problems 409 11.13 It was stated that the inclusion of the salt in the UNIX password scheme increases the difficulty of guessing by a factor of 4096. But the salt is stored in plaintext in the same entry as the corresponding ciphertext password. Therefore, those two characters are known to the attacker and need not be guessed. Why is it asserted that the salt increases security? 11.14 Assuming that you have successfully answered the preceding problem and understand the significance of the salt, here is another question. Wouldn't it be possible to thwart completely all password crackers by dramatically increasing the salt size to, say, 24 or 48 bits? 11.15 Consider the Bloom filter discussed in Section 11.3. Define k = number of hash functions; N = number of bits in hash table; and D = number of words in dictionary. a. Show that the expected number of bits in the hash table that are equal to zero is expressed as f = a1 - k Nb D b. Show that the probability that an input word, not in the dictionary, will be falsely accepted as being in the dictionary is P = (1 - f) k c. Show that the preceding expression can be approximated as P ≈ (1 - e-kD/N) k 11.16 Design a file access system to allow certain users read and write

access to files, depending on authorization set up by the system. The instructions should be of the format: ReadFile(F1, User A): User A has read access to file F1 WriteFile(F2, User A): User A has write access to file F2 ExecuteFile(F3, User B): User B has execute access to file F3 Each file has a header record, which contains authorization privileges; that is, a list of users who can read and write. The file is to be encrypted by a key that is not shared by the users but known only to the system. 410 12.1 The Need for Firewalls 12.2 Firewall Characteristics and Access Policy 12.3 Types of Firewalls Packet Filtering Firewall Stateful Inspection Firewalls Application-Level Gateway Circuit-Level Gateway 12.4 Firewall Basing Bastion Host Host-Based Firewalls Personal Firewall 12.5 Firewall Location and Configurations DMZ Networks Virtual Private Networks Distributed Firewalls Summary of Firewall Locations and Topologies 12.6 Key Terms, Review Questions, and Problems Chapter Firewalls 12.1 / The Need for Firewalls 411 Firewalls can be an effective means of protecting a local system or network of systems from network-based security threats while at the same time affording access to the outside world via wide area networks and the Internet. 12.1 The Need for Firewalls Information systems in corporations, government agencies, and other organizations have undergone a steady evolution. The following are notable developments: ■■ Centralized data processing system, with a central mainframe supporting a number of directly connected terminals ■■ Local area networks (LANs) interconnecting PCs and terminals to each other and the mainframe ■■ Premises network, consisting of a number of LANs, interconnecting PCs, servers, and perhaps a mainframe or two ■■ Enterprise-wide network, consisting of multiple, geographically distributed premises networks interconnected by a private wide area network (WAN) ■■ Internet connectivity, in which the various premises networks all hook into the Internet and may or may not also be connected by a private WAN Internet connectivity is no longer optional for organizations. The information and services available are essential to the organization. Moreover, individual users within the organization want and need Internet access, and if this is not provided via their LAN, they will use dial-up capability from their PC to an Internet service provider (ISP). However, while Internet access provides benefits to the organization, it enables the outside world to reach and interact with local network assets. This creates a threat to the organization. While it is possible to equip each workstation and server on the premises network with strong security features, such as intrusion protection, this may not be sufficient and in some cases is not cost-effective. Consider a network with hundreds or even thousands of systems, running various operating systems, such as different versions of UNIX and Windows. When a security flaw Learning Objectives After studying this chapter, you should be able to: ◆ Explain the role of firewalls as part of a computer and network security strategy. ◆ List the key characteristics of firewalls. ◆ Discuss the various basing options for firewalls. ◆ Understand the relative merits of various choices for firewall location and configurations. 412 chapter 12 / Firewalls is discovered, each potentially affected system must be upgraded to fix that flaw. This requires scaleable configuration management and aggressive patching to function effectively. While difficult, this is possible and is necessary if only host-based security is used. A widely accepted alternative or at least complement to host-based security services is the firewall. The firewall is inserted between the premises network and the Internet to establish a controlled link and to erect an outer security wall or perimeter. The aim of this perimeter is to protect the premises network from Internet-based attacks and to provide a single choke point where security and auditing can be imposed. The firewall may be a single computer system or a set of two or more systems that cooperate to perform the firewall function. The firewall, then, provides an additional layer of defense,

insulating the internal systems from external networks. This follows the classic military doctrine of “defense in depth,” which is just as applicable to IT security. 12.2 Firewall Characteristics and Access Policy [BELL94b] lists the following design goals for a firewall: 1. All traffic from inside to outside, and vice versa, must pass through the firewall. This is achieved by physically blocking all access to the local network except via the firewall. Various configurations are possible, as explained later in this chapter. 2. Only authorized traffic, as defined by the local security policy, will be allowed to pass. Various types of firewalls are used, which implement various types of security policies, as explained later in this chapter. 3. The firewall itself is immune to penetration. This implies the use of a hardened system with a secured operating system. Trusted computer systems are suitable for hosting a firewall and often required in government applications. A critical component in the planning and implementation of a firewall is specifying a suitable access policy. This lists the types of traffic authorized to pass through the firewall, including address ranges, protocols, applications, and content types. This policy should be developed from the organization’s information security risk assessment and policy. This policy should be developed from a broad specification of which traffic types the organization needs to support. It is then refined to detail the filter elements we discuss next, which can then be implemented within an appropriate firewall topology. SP 800-41-1 (Guidelines on Firewalls and Firewall Policy, September 2009) lists a range of characteristics that a firewall access policy could use to filter traffic, including: ■■ IP Address and Protocol Values: Controls access based on the source or destination addresses and port numbers, direction of flow being inbound or outbound, and other network and transport layer characteristics. This type of filtering is used by packet filter and stateful inspection firewalls. It is typically used to limit access to specific services. 12.2 / Firewall Characteristics and Access Policy 413 ■■ Application Protocol: Controls access on the basis of authorized application protocol data. This type of filtering is used by an application-level gateway that relays and monitors the exchange of information for specific application protocols, for example, checking SMTP e-mail for spam, or HTPP Web requests to authorized sites only. ■■ User Identity: Controls access based on the users identity, typically for inside users who identify themselves using some form of secure authentication technology, such as IPSec (Chapter 9). ■■ Network Activity: Controls access based on considerations such as the time or request, for example, only in business hours; rate of requests, for example, to detect scanning attempts; or other activity patterns. Before proceeding to the details of firewall types and configurations, it is best to summarize what one can expect from a firewall. The following capabilities are within the scope of a firewall: 1. A firewall defines a single choke point that keeps unauthorized users out of the protected network, prohibits potentially vulnerable services from entering or leaving the network, and provides protection from various kinds of IP spoofing and routing attacks. The use of a single choke point simplifies security management because security capabilities are consolidated on a single system or set of systems. 2. A firewall provides a location for monitoring security-related events. Audits and alarms can be implemented on the firewall system. 3. A firewall is a convenient platform for several Internet functions that are not security related. These include a network address translator, which maps local addresses to Internet addresses, and a network management function that audits or logs Internet usage. 4. A firewall can serve as the platform for IPsec. Using the tunnel mode capability described in Chapter 9, the firewall can be used to implement virtual private networks. Firewalls have their limitations, including the following: 1. The firewall cannot protect against attacks that bypass the firewall. Internal systems may have dial-

out capability to connect to an ISP. An internal LAN may support a modem pool that provides dial-in capability for traveling employees and telecommuters. 2. The firewall may not protect fully against internal threats, such as a disgruntled employee or an employee who unwittingly cooperates with an external attacker. 3. An improperly secured wireless LAN may be accessed from outside the organization. An internal firewall that separates portions of an enterprise network cannot guard against wireless communications between local systems on different sides of the internal firewall. 4. A laptop, PDA, or portable storage device may be used and infected outside the corporate network, and then attached and used internally. 414 chapter 12 / Firewalls 12.3 Types of Firewalls A firewall can monitor network traffic at a number of levels, from low-level network packets either individually or as part of a flow, to all traffic within a transport connection, up to inspecting details of application protocols. The choice of which level is appropriate is determined by the desired firewall access policy. It can operate as a positive filter, allowing to pass only packets that meet specific criteria, or as a negative filter, rejecting any packet that meets certain criteria. The criteria implement the access policy for the firewall, that we discussed in the previous section. Depending on the type of firewall, it may examine one or more protocol headers in each packet, the payload of each packet, or the pattern generated by a sequence of packets. In this section, we look at the principal types of firewalls. Packet Filtering Firewall A packet filtering firewall applies a set of rules to each incoming and outgoing IP packet and then forwards or discards the packet (Figure 12.1b). The firewall is typically configured to filter packets going in both directions (from and to the internal network). Filtering rules are based on information contained in a network packet: ■■ Source IP address: The IP address of the system that originated the IP packet (e.g., 192.178.1.1) ■■ Destination IP address: The IP address of the system the IP packet is trying to reach (e.g., 192.168.1.2) ■■ Source and destination transport-level address: The transport-level (e.g., TCP or UDP) port number, which defines applications such as SNMP or TELNET ■■ IP protocol field: Defines the transport protocol ■■ Interface: For a firewall with three or more ports, which interface of the firewall the packet came from or which interface of the firewall the packet is destined for The packet filter is typically set up as a list of rules based on matches to fields in the IP or TCP header. If there is a match to one of the rules, that rule is invoked to determine whether to forward or discard the packet. If there is no match to any rule, then a default action is taken. Two default policies are possible: ■■ Default = discard: That which is not expressly permitted is prohibited. ■■ Default = forward: That which is not expressly prohibited is permitted. The default discard policy is more conservative. Initially, everything is blocked, and services must be added on a case-by-case basis. This policy is more visible to users, who are more likely to see the firewall as a hindrance. However, this is the policy likely to be preferred by businesses and government organizations. Further, visibility to users diminishes as rules are created. The default forward policy increases ease of use for end users but provides reduced security; the security administrator must, in essence, react to each new security threat as it becomes known. This policy may be used by generally more open organizations, such as universities. 12.3 / Types of Firewalls 415 Figure 12.1 Types of Firewalls External (untrusted) network (e.g., Internet) Internal (protected) network (e.g., enterprise network) Firewall (a) General model (d) Application proxy rewall Physical Network access Internet Transport Application Physical Network access Internet Transport Application Application proxy External transport connection Internal transport connection (b) Packet ltering rewall Physical Network access Internet Transport Application End-to-end transport connection End-to-end transport connection (c) Stateful inspection rewall Physical Network access

Internet Transport Application End-to-end transport connection End-to-end transport connection (e) Circuit-level proxy rewall Physical Network access Internet Transport Application Physical Network access Internet Transport Application Circuit-level proxy External transport connection Internal transport connection State info Table 12.1 is a simplified example of a ruleset for SMTP traffic. The goal is to allow inbound and outbound e-mail traffic but to block all other traffic. The rules are applied top to bottom to each packet. M12_S 416 chapter 12 / Firewalls A. Inbound mail from an external source is allowed (port 25 is for SMTP incoming). B. This rule is intended to allow a response to an inbound SMTP connection. C. Outbound mail to an external source is allowed. D. This rule is intended to allow a response to an inbound SMTP connection. E. This is an explicit statement of the default policy. All rulesets include this rule implicitly as the last rule. There are several problems with this ruleset. Rule D allows external traffic to any destination port above 1023. As an example of an exploit of this rule, an external attacker can open a connection from the attacker's port 5150 to an internal Web proxy server on port 8080. This is supposed to be forbidden and could allow an attack on the server. To counter this attack, the firewall ruleset can be configured with a source port field for each row. For rules B and D, the source port is set to 25; for rules A and C, the source port is set to 7 1023. But a vulnerability remains. Rules C and D are intended to specify that any inside host can send mail to the outside. A TCP packet with a destination port of 25 is routed to the SMTP server on the destination machine. The problem with this rule is that the use of port 25 for SMTP receipt is only a default; an outside machine could be configured to have some other application linked to port 25. As the revised rule D is written, an attacker could gain access to internal machines by sending packets with a TCP source port number of 25. To counter this threat, we can add an ACK flag field to each row. For rule D, the field would indicate that the ACK flag must be set on the incoming packet. Rule D would now look like this: Rule Direction Source Address Source Port Dest Address Protocol Dest Port Flag Action D In External 25 Internal TCP 7 1023 ACK Permit The rule takes advantage of a feature of TCP connections. Once a connection is set up, the ACK flag of a TCP segment is set to acknowledge segments sent from the other side. Thus, this rule allows incoming packets with a source port number of 25 that include the ACK flag in the TCP segment. Rule Direction Source Address Destination Address Protocol Destination Port Action A In External Internal TCP 25 Permit B Out Internal External TCP 7 1023 Permit C Out Internal External TCP 25 Permit D In External Internal TCP 7 1023 Permit E Either Any Any Any Any Deny Table 12.1 Packet-Filtering Example 12.3 / Types of Firewalls 417 One advantage of a packet filtering firewall is its simplicity. Also, packet filters typically are transparent to users and are very fast. [SP 800-41-1] lists the following weaknesses of packet filter firewalls: ■■ Because packet filter firewalls do not examine upper-layer data, they cannot prevent attacks that employ application-specific vulnerabilities or functions. For example, a packet filter firewall cannot block specific application commands; if a packet filter firewall allows a given application, all functions available within that application will be permitted. ■■ Because of the limited information available to the firewall, the logging functionality present in packet filter firewalls is limited. Packet filter logs normally contain the same information used to make access control decisions (source address, destination address, and traffic type). ■■ Most packet filter firewalls do not support advanced user authentication schemes. Once again, this limitation is mostly due to the lack of upper-layer functionality by the firewall. ■■ Packet filter firewalls are generally vulnerable to attacks and exploits that take advantage of problems within the TCP/IP specification and protocol stack, such as network layer address spoofing. Many packet filter firewalls

cannot detect a network packet in which the OSI Layer 3 addressing information has been altered. Spoofing attacks are generally employed by intruders to bypass the security controls implemented in a firewall platform. ■■ Finally, due to the small number of variables used in access control decisions, packet filter firewalls are susceptible to security breaches caused by improper configurations. In other words, it is easy to accidentally configure a packet filter firewall to allow traffic types, sources, and destinations that should be denied based on an organization's information security policy. Some of the attacks that can be made on packet filtering firewalls and the appropriate countermeasures are the following: ■■ IP address spoofing: The intruder transmits packets from the outside with a source IP address field containing an address of an internal host. The attacker hopes that the use of a spoofed address will allow penetration of systems that employ simple source address security, in which packets from specific trusted internal hosts are accepted. The countermeasure is to discard packets with an inside source address if the packet arrives on an external interface. In fact, this countermeasure is often implemented at the router external to the firewall. ■■ Source routing attacks: The source station specifies the route that a packet should take as it crosses the Internet, in the hopes that this will bypass security measures that do not analyze the source routing information. The countermeasure is to discard all packets that use this option. ■■ Tiny fragment attacks: The intruder uses the IP fragmentation option to create extremely small fragments and force the TCP header information into a separate packet fragment. This attack is designed to circumvent filtering rules that depend on TCP header information. Typically, a packet filter will make a filtering decision on the first fragment of a packet. All 418 chapter 12 / Firewalls subsequent fragments of that packet are filtered out solely on the basis that they are part of the packet whose first fragment was rejected. The attacker hopes that the filtering firewall examines only the first fragment and that the remaining fragments are passed through. A tiny fragment attack can be defeated by enforcing a rule that the first fragment of a packet must contain a predefined minimum amount of the transport header. If the first fragment is rejected, the filter can remember the packet and discard all subsequent fragments. Stateful Inspection Firewalls A traditional packet filter makes filtering decisions on an individual packet basis and does not take into consideration any higher-layer context. To understand what is meant by context and why a traditional packet filter is limited with regard to context, a little background is needed. Most standardized applications that run on top of TCP follow a client/server model. For example, for the Simple Mail Transfer Protocol (SMTP), e-mail is transmitted from a client system to a server system. The client system generates new e-mail messages, typically from user input. The server system accepts incoming e-mail messages and places them in the appropriate user mailboxes. SMTP operates by setting up a TCP connection between client and server, in which the TCP server port number, which identifies the SMTP server application, is 25. The TCP port number for the SMTP client is a number between 1024 and 65535 that is generated by the SMTP client. In general, when an application that uses TCP creates a session with a remote host, it creates a TCP connection in which the TCP port number for the remote (server) application is a number less than 1024 and the TCP port number for the local (client) application is a number between 1024 and 65535. The numbers less than 1024 are the "well-known" port numbers and are assigned permanently to particular applications (e.g., 25 for server SMTP). The numbers between 1024 and 65535 are generated dynamically and have temporary significance only for the lifetime of a TCP connection. A simple packet filtering firewall must permit inbound network traffic on all these high-numbered ports for TCP-

based traffic to occur. This creates a vulnerability that can be exploited by unauthorized users. A stateful inspection packet firewall tightens up the rules for TCP traffic by creating a directory of outbound TCP connections, as shown in Table 12.2. There is an entry for each currently established connection. The packet filter will now allow incoming traffic to high-numbered ports only for those packets that fit the profile of one of the entries in this directory. A stateful packet inspection firewall reviews the same packet information as a packet filtering firewall, but also records information about TCP connections (Figure 12.1c). Some stateful firewalls also keep track of TCP sequence numbers to prevent attacks that depend on the sequence number, such as session hijacking. Some even inspect limited amounts of application data for some well-known protocols like FTP, IM and SIPS commands, in order to identify and track related connections. 12.3 / Types of Firewalls 419 Application-Level Gateway An application-level gateway, also called an application proxy, acts as a relay of application-level traffic (Figure 12.1d). The user contacts the gateway using a TCP/ IP application, such as Telnet or FTP, and the gateway asks the user for the name of the remote host to be accessed. When the user responds and provides a valid user ID and authentication information, the gateway contacts the application on the remote host and relays TCP segments containing the application data between the two endpoints. If the gateway does not implement the proxy code for a specific application, the service is not supported and cannot be forwarded across the firewall. Further, the gateway can be configured to support only specific features of an application that the network administrator considers acceptable while denying all other features. Application-level gateways tend to be more secure than packet filters. Rather than trying to deal with the numerous possible combinations that are to be allowed and forbidden at the TCP and IP level, the application-level gateway need only scrutinize a few allowable applications. In addition, it is easy to log and audit all incoming traffic at the application level. A prime disadvantage of this type of gateway is the additional processing overhead on each connection. In effect, there are two spliced connections between the end users, with the gateway at the splice point, and the gateway must examine and forward all traffic in both directions. Circuit-Level Gateway A fourth type of firewall is the circuit-level gateway or circuit-level proxy (Figure 12.1e). This can be a stand-alone system or it can be a specialized function performed by an application-level gateway for certain applications. As with an Source Address Source Port Destination Address Destination Port Connection State 192.168.1.100 1030 210.22.88.29 80 Established 192.168.1.102 1031 216.32.42.123 80 Established 192.168.1.101 1033 173.66.32.122 25 Established 192.168.1.106 1035 177.231.32.12 79 Established 223.43.21.231 1990 192.168.1.6 80 Established 2122.22.123.32 2112 192.168.1.6 80 Established 210.922.212.18 3321 192.168.1.6 80 Established 24.102.32.23 1025 192.168.1.6 80 Established 223.21.22.12 1046 192.168.1.6 80 Established Table 12.2 Example Stateful Firewall Connection State Table (SP 800-41-1) 420 chapter 12 / Firewalls application gateway, a circuit-level gateway does not permit an end-to-end TCP connection; rather, the gateway sets up two TCP connections, one between itself and a TCP user on an inner host and one between itself and a TCP user on an outside host. Once the two connections are established, the gateway typically relays TCP segments from one connection to the other without examining the contents. The security function consists of determining which connections will be allowed. A typical use of circuit-level gateways is a situation in which the system administrator trusts the internal users. The gateway can be configured to support application-level or proxy service on inbound connections and circuit-level functions for outbound connections. In this configuration, the gateway can incur the processing

overhead of examining incoming application data for forbidden functions but does not incur that overhead on outgoing data. An example of a circuit-level gateway implementation is the SOCKS package [KOBL92]; version 5 of SOCKS is specified in RFC 1928. The SOCKS protocol provides a framework for client-server applications in both the TCP and UDP domains. It is designed to provide convenient and secure access to a network-level firewall. The protocol occupies a thin layer between the application and either TCP or UDP but does not provide network-level routing services, such as forwarding of ICMP messages. SOCKS consists of the following components: ■■ The SOCKS server, which often runs on a UNIX-based firewall. SOCKS is also implemented on Windows systems. ■■ The SOCKS client library, which runs on internal hosts protected by the firewall. ■■ SOCKS-ified versions of several standard client programs such as FTP and TELNET. The implementation of the SOCKS protocol typically involves either the recompilation or relinking of TCP-based client applications or the use of alternate dynamically loaded libraries, to use the appropriate encapsulation routines in the SOCKS library. When a TCP-based client wishes to establish a connection to an object that is reachable only via a firewall (such determination is left up to the implementation), it must open a TCP connection to the appropriate SOCKS port on the SOCKS server system. The SOCKS service is located on TCP port 1080. If the connection request succeeds, the client enters a negotiation for the authentication method to be used, authenticates with the chosen method, and then sends a relay request. The SOCKS server evaluates the request and either establishes the appropriate connection or denies it. UDP exchanges are handled in a similar fashion. In essence, a TCP connection is opened to authenticate a user to send and receive UDP segments, and the UDP segments are forwarded as long as the TCP connection is open. 12.4 Firewall Basing It is common to base a firewall on a stand-alone machine running a common operating system, such as UNIX or Linux. Firewall functionality can also be implemented as a software module in a router or LAN switch. In this section, we look at some additional firewall basing considerations. 12.4 / Firewall Basing 421 Bastion Host A bastion host is a system identified by the firewall administrator as a critical strong point in the network's security. Typically, the bastion host serves as a platform for an application-level or circuit-level gateway. Common characteristics of a bastion host are as follows: ■■ The bastion host hardware platform executes a secure version of its operating system, making it a hardened system. ■■ Only the services that the network administrator considers essential are installed on the bastion host. These could include proxy applications for DNS, FTP, HTTP, and SMTP. ■■ The bastion host may require additional authentication before a user is allowed access to the proxy services. In addition, each proxy service may require its own authentication before granting user access. ■■ Each proxy is configured to support only a subset of the standard application's command set. ■■ Each proxy is configured to allow access only to specific host systems. This means that the limited command/feature set may be applied only to a subset of systems on the protected network. ■■ Each proxy maintains detailed audit information by logging all traffic, each connection, and the duration of each connection. The audit log is an essential tool for discovering and terminating intruder attacks. ■■ Each proxy module is a very small software package specifically designed for network security. Because of its relative simplicity, it is easier to check such modules for security flaws. For example, a typical UNIX mail application may contain over 20,000 lines of code, while a mail proxy may contain fewer than 1000. ■■ Each proxy is independent of other proxies on the bastion host. If there is a problem with the operation of any proxy, or if a future vulnerability is discovered, it can be uninstalled without affecting the operation of the other proxy

applications. Also, if the user population requires support for a new service, the network administrator can easily install the required proxy on the bastion host. ■■ A proxy generally performs no disk access other than to read its initial configuration file. Hence, the portions of the file system containing executable code can be made read only. This makes it difficult for an intruder to install Trojan horse sniffers or other dangerous files on the bastion host. ■■ Each proxy runs as a nonprivileged user in a private and secured directory on the bastion host. Host-Based Firewalls A host-based firewall is a software module used to secure an individual host. Such modules are available in many operating systems or can be provided as an addon package. Like conventional stand-alone firewalls, host-resident firewalls filter 422 chapter 12 / Firewalls and restrict the flow of packets. A common location for such firewalls is a server. There are several advantages to the use of a server-based or workstation-based firewall: ■■ Filtering rules can be tailored to the host environment. Specific corporate security policies for servers can be implemented, with different filters for servers used for different application. ■■ Protection is provided independent of topology. Thus both internal and external attacks must pass through the firewall. ■■ Used in conjunction with stand-alone firewalls, the host-based firewall provides an additional layer of protection. A new type of server can be added to the network, with its own firewall, without the necessity of altering the network firewall configuration. Personal Firewall A personal firewall controls the traffic between a personal computer or workstation on one side and the Internet or enterprise network on the other side. Personal firewall functionality can be used in the home environment and on corporate intranets. Typically, the personal firewall is a software module on the personal computer. In a home environment with multiple computers connected to the Internet, firewall functionality can also be housed in a router that connects all of the home computers to a DSL, cable modem, or other Internet interface. Personal firewalls are typically much less complex than either server-based firewalls or stand-alone firewalls. The primary role of the personal firewall is to deny unauthorized remote access to the computer. The firewall can also monitor outgoing activity in an attempt to detect and block worms and other malware. Personal firewall capabilities are provided by the netfilter package on Linux systems, or the pf package on BSD and Mac OS X systems. These packages may be configured on the command-line, or with a GUI front-end. When such a personal firewall is enabled, all inbound connections are usually denied except for those the user explicitly permits. Outbound connections are usually allowed. The list of inbound services that can be selectively re-enabled, with their port numbers, may include the following common services: ■■ Personal file sharing (548, 427) ■■ Windows sharing (139) ■■ Personal Web sharing (80, 427) ■■ Remote login—SSH (22) ■■ FTP access (20–21, 1024-65535 from 20–21) ■■ Printer sharing (631, 515) ■■ iChat Rendezvous (5297, 5298) ■■ iTunes Music Sharing (3869) ■■ CVS (2401) ■■ Gnutella/Limewire (6346) 12.5 / Firewall Location and Configurations 423 ■■ ICQ (4000) ■■ IRC (194) ■■ MSN Messenger (6891–6900) ■■ Network Time (123) ■■ Retrospect (497) ■■ SMB (without netbios; 445) ■■ Timbuktu (407) ■■ VNC (5900–5902) ■■ WebSTAR Admin (1080, 1443) When FTP access is enabled, ports 20 and 21 on the local machine are opened for FTP; if others connect this computer from ports 20 or 21, the ports 1024 through 65535 are open. For increased protection, advanced firewall features may be configured. For example, stealth mode hides the system on the Internet by dropping unsolicited communication packets, making it appear as though the system is not present. UDP packets can be blocked, restricting network traffic to TCP packets only for open ports. The firewall also supports logging, an important tool for checking on unwanted activity. Other types of personal firewall allow the user to specify

that only selected applications, or applications signed by a valid certificate authority, may provide services accessed from the network. 12.5 Firewall Location and Configurations As Figure 12.1a indicates, a firewall is positioned to provide a protective barrier between an external, potentially untrusted source of traffic and an internal network. With that general principle in mind, a security administrator must decide on the location and on the number of firewalls needed. In this section, we look at some common options. DMZ Networks Figure 12.2 suggests the most common distinction, that between an internal and an external firewall. An external firewall is placed at the edge of a local or enterprise network, just inside the boundary router that connects to the Internet or some wide area network (WAN). One or more internal firewalls protect the bulk of the enterprise network. Between these two types of firewalls are one or more networked devices in a region referred to as a DMZ (demilitarized zone) network. Systems that are externally accessible but need some protections are usually located on DMZ networks. Typically, the systems in the DMZ require or foster external connectivity, such as a corporate Web site, an e-mail server, or a DNS (domain name system) server. The external firewall provides a measure of access control and protection for the DMZ systems consistent with their need for external connectivity. The external 424 chapter 12 / Firewalls firewall also provides a basic level of protection for the remainder of the enterprise network. In this type of configuration, internal firewalls serve three purposes: 1. The internal firewall adds more stringent filtering capability, compared to the external firewall, in order to protect enterprise servers and workstations from external attack. 2. The internal firewall provides two-way protection with respect to the DMZ. First, the internal firewall protects the remainder of the network from attacks launched from DMZ systems. Such attacks might originate from worms, rootkits, bots, or other malware lodged in a DMZ system. Second, an internal firewall can protect the DMZ systems from attack from the internal protected network. Figure 12.2 Example Firewall Configuration Workstations Application and database servers Web server(s) E-mail server Internal DMZ network Boundary router External •rewall LAN switch LAN switch Internal •rewall Internal protected network DNS server Remote Internet users 12.5 / Firewall Location and Configurations 425 3. Multiple internal firewalls can be used to protect portions of the internal network from each other. For example, firewalls can be configured so that internal servers are protected from internal workstations and vice versa. A common practice is to place the DMZ on a different network interface on the external firewall from that used to access the internal networks. Virtual Private Networks In today's distributed computing environment, the virtual private network (VPN) offers an attractive solution to network managers. In essence, a VPN consists of a set of computers that interconnect by means of a relatively unsecure network and that make use of encryption and special protocols to provide security. At each corporate site, workstations, servers, and databases are linked by one or more local area networks (LANs). The Internet or some other public network can be used to interconnect sites, providing a cost savings over the use of a private network and offloading the wide area network management task to the public network provider. That same public network provides an access path for telecommuters and other mobile employees to log on to corporate systems from remote sites. But the manager faces a fundamental requirement: security. Use of a public network exposes corporate traffic to eavesdropping and provides an entry point for unauthorized users. To counter this problem, a VPN is needed. In essence, a VPN uses encryption and authentication in the lower protocol layers to provide a secure connection through an otherwise insecure network, typically the Internet. VPNs are generally cheaper than real private networks

using private lines but rely on having the same encryption and authentication system at both ends. The encryption may be performed by firewall software or possibly by routers. The most common protocol mechanism used for this purpose is at the IP level and is known as IPsec. Figure 12.3 (Compare Figure 9.1) is a typical scenario of IPSec usage.1 An organization maintains LANs at dispersed locations. Nonsecure IP traffic is conducted on each LAN. For traffic off site, through some sort of private or public WAN, IPSec protocols are used. These protocols operate in networking devices, such as a router or firewall, that connect each LAN to the outside world. The IPSec networking device will typically encrypt and compress all traffic going into the WAN and decrypt and uncompress traffic coming from the WAN; authentication may also be provided. These operations are transparent to workstations and servers on the LAN. Secure transmission is also possible with individual users who dial into the WAN. Such user workstations must implement the IPSec protocols to provide security. They must also implement high levels of host security, as they are directly connected to the wider Internet. This makes them an attractive target for attackers attempting to access the corporate network. A logical means of implementing an IPSec is in a firewall, as shown in Figure 12.3. If IPSec is implemented in a separate box behind (internal to) the firewall, then VPN traffic passing through the firewall in both directions is encrypted. In this case, the firewall is unable to perform its filtering function or other security 1 Details of IPSec are provided in Chapter 9. For this discussion, all that we need to know is that IPSec adds one or more additional headers to the IP packet to support encryption and authentication functions. 426 chapter 12 / Firewalls functions, such as access control, logging, or scanning for viruses. IPSec could be implemented in the boundary router, outside the firewall. However, this device is likely to be less secure than the firewall and thus less desirable as an IPSec platform. Distributed Firewalls A distributed firewall configuration involves stand-alone firewall devices plus hostbased firewalls working together under a central administrative control. Figure 12.4 suggests a distributed firewall configuration. Administrators can configure hostresident firewalls on hundreds of servers and workstations as well as configure personal firewalls on local and remote user systems. Tools let the network administrator set policies and monitor security across the entire network. These firewalls protect against internal attacks and provide protection tailored to specific machines and applications. Stand-alone firewalls provide global protection, including internal firewalls and an external firewall, as discussed previously. With distributed firewalls, it may make sense to establish both an internal and an external DMZ. Web servers that need less protection because they have less critical information on them could be placed in an external DMZ, outside the external firewall. What protection is needed is provided by host-based firewalls on these servers. An important aspect of a distributed firewall configuration is security monitoring. Such monitoring typically includes log aggregation and analysis, firewall statistics, and fine-grained remote monitoring of individual hosts if needed. Figure 12.3 A VPN Security Scenario IP header IP payload IP header IPSec header Secure IP payload header IP IPSec header Secure IP payload IP header IPSec header Secure IP payload IP header IP payload Firewall with IPSec Ethernet switch Ethernet switch User system with IPSec Firewall with IPSec Public (Internet) or private network 12.5 / Firewall Location and Configurations 427 Summary of Firewall Locations and Topologies We can now summarize the discussion from Sections 12.4 and 12.5 to define a spectrum of firewall locations and topologies. The following alternatives can be identified: ■■ Host-resident firewall: This category includes personal firewall software and firewall software on servers. Such firewalls can be used alone or as part of an in-depth firewall deployment. Figure 12.4 Example Distributed

Firewall Configuration Workstations Application and database servers Web server(s) E-mail server Internal DMZ network Boundary router External •rewall LAN switch LAN switch host-resident •rewall Internal •rewall Internal protected network DNS server Internet Web server(s) External DMZ network Remote users 428 chapter 12 / Firewalls ■■ Screening router: A single router between internal and external networks with stateless or full packet filtering. This arrangement is typical for small office/home office (SOHO) applications. ■■ Single bastion inline: A single firewall device between an internal and external router (e.g., Figure 12.1a). The firewall may implement stateful filters and/ or application proxies. This is the typical firewall appliance configuration for small- to medium-sized organizations. ■■ Single bastion T: Similar to single bastion inline but has a third network interface on bastion to a DMZ where externally visible servers are placed. Again, this is a common appliance configuration for medium to large organizations. ■■ Double bastion inline: Figure 12.2 illustrates this configuration, where the DMZ is sandwiched between bastion firewalls. This configuration is common for large businesses and government organizations. ■■ Double bastion T: The DMZ is on a separate network interface on the bastion firewall. This configuration is also common for large businesses and government organizations and may be required. For example, this configuration is required for Australian government use (Australian Government Information Technology Security Manual—ACSI33). ■■ Distributed firewall configuration: Illustrated in Figure 12.4. This configuration is used by some large businesses and government organizations. 12.6 Key Terms, Review Questions, and Problems Key Terms application-level gateway bastion host circuit-level gateway distributed firewalls DMZ firewall host-based firewall IP address spoofing IP security (IPSec) packet filtering firewall personal firewall proxy stateful inspection firewall tiny fragment attack virtual private network (VPN) Review Questions 12.1 List three design goals for a firewall. 12.2 List four techniques used by firewalls to control access and enforce a security policy. 12.3 When does a packet filtering firewall resort to default actions? List these default policies. 12.4 What are some weaknesses of a packet filtering firewall? 12.5 Explain three attacks that can be made on packet filtering firewalls. What measures can be taken to counter these attacks? 12.6 What is an application-level gateway? 12.7 What is a circuit-level gateway? 12.8 What are the differences among the firewalls of Figure 12.1? 12.9 What are the common characteristics of a bastion host? 12.6 / Key Terms, Review Questions, and Problems 429 12.10 Why is it useful to have host-based firewalls? 12.11 What is a virtual private network? How does it ensure a secure connection? 12.12 Describe the spectrum of firewall locations and topologies. Problems 12.1 As was mentioned in Section 12.3, one approach to defeating the tiny fragment attack is to enforce a minimum length of the transport header that must be contained in the first fragment of an IP packet. If the first fragment is rejected, all subsequent fragments can be rejected. However, the nature of IP is such that fragments may arrive out of order. Thus, an intermediate fragment may pass through the filter before the initial fragment is rejected. How can this situation be handled? 12.2 In an IPv4 packet, the size of the payload in the first fragment, in octets, is equal to Total Length - (4 * IHL). If this value is less than the required minimum (8 octets for TCP), then this fragment and the entire packet are rejected. Suggest an alternative method of achieving the same result using only the Fragment Offset field. 12.3 RFC 791, the IPv4 protocol specification, describes a reassembly algorithm that results in new fragments overwriting any overlapped portions of previously received fragments. Given such a reassembly implementation, an attacker could construct a series of packets in which the lowest (zero-offset) fragment would contain innocuous data (and thereby be passed by administrative

packet filters), and in which some subsequent packet having a non-zero offset would overlap TCP header information (destination port, for instance) and cause it to be modified. The second packet would be passed through most filter implementations because it does not have a zero fragment offset. Suggest a method that could be used by a packet filter to counter this attack. 12.4 Table 12.3 shows a sample of a packet filter firewall ruleset for an imaginary network of IP address that range from 192.168.1.0 to 192.168.1.254. Describe the effect of each rule. 12.5 SMTP (Simple Mail Transfer Protocol) is the standard protocol for transferring mail between hosts over TCP. A TCP connection is set up between a user agent and a server program. The server listens on TCP port 25 for incoming connection requests. The user end of the connection is on a TCP port number above 1023. Suppose you wish to build a packet filter ruleset allowing inbound and outbound SMTP traffic. You generate the following ruleset: Source Address Source Port Dest Address Dest Port Action 1 Any Any 192.168.1.0 7 1023 Allow 2 192.168.1.1 Any Any Any Deny 3 Any Any 192.168.1.1 Any Deny 4 192.168.1.0 Any Any Any Allow 5 Any Any 192.168.1.2 SMTP Allow 6 Any Any 192.168.1.3 HTTP Allow 7 Any Any Any Any Deny Table 12.3 Sample Packet Filter Firewall Ruleset 430 chapter 12 / Firewalls Rule Direction Src Addr Dest Addr Protocol Dest Port Action A In External Internal TCP 25 Permit B Out Internal External TCP 71023 Permit C Out Internal External TCP 25 Permit D In External Internal TCP 71023 Permit E Either Any Any Any Any Deny a. Describe the effect of each rule. b. Your host in this example has IP address 172.16.1.1. Someone tries to send e-mail from a remote host with IP address 192.168.3.4. If successful, this generates an SMTP dialogue between the remote user and the SMTP server on your host consisting of SMTP commands and mail. Additionally, assume that a user on your host tries to send e-mail to the SMTP server on the remote system. Four typical packets for this scenario are as shown: Packet Direction Src Addr Dest Addr Protocol Dest Port Action 1 In 192.168.3.4 172.16.1.1 TCP 25 ? 2 Out 172.16.1.1 192.168.3.4 TCP 1234 ? 3 Out 172.16.1.1 192.168.3.4 TCP 25 ? 4 In 192.168.3.4 172.16.1.1 TCP 1357 ? Indicate which packets are permitted or denied and which rule is used in each case. c. Someone from the outside world (10.1.2.3) attempts to open a connection from port 5150 on a remote host to the Web proxy server on port 8080 on one of your local hosts (172.16.3.4), in order to carry out an attack. Typical packets are as follows: Packet Direction Src Addr Dest Addr Protocol Dest Port Action 5 In 10.1.2.3 172.16.3.4 TCP 8080 ? 6 Out 172.16.3.4 10.1.2.3 TCP 5150 ? Will the attack succeed? Give details. 12.6 To provide more protection, the ruleset from the preceding problem is modified as follows: Rule Direction Src Addr Dest Addr Protocol Src Port Dest Port Action A In External Internal TCP 71023 25 Permit B Out Internal External TCP 25 71023 Permit C Out Internal External TCP 71023 25 Permit D In External Internal TCP 25 71023 Permit E Either Any Any Any Any Any Deny a. Describe the change. b. Apply this new ruleset to the same six packets of the preceding problem. Indicate which packets are permitted or denied and which rule is used in each case. 12.6 / Key Terms, Review Questions, and Problems 431 12.7 A hacker uses port 25 as the client port on his or her end to attempt to open a connection to your Web proxy server. a. The following packets might be generated: Packet Direction Src Addr Dest Addr Protocol Src Port Dest Port Action 7 In 10.1.2.3 172.16.3.4 TCP 25 8080 ? 8 Out 172.16.3.4 10.1.2.3 TCP 8080 25 ? Explain why this attack will succeed, using the ruleset of the preceding problem. b. When a TCP connection is initiated, the ACK bit in the TCP header is not set. Subsequently, all TCP headers sent over the TCP connection have the ACK bit set. Use this information to modify the ruleset of the preceding problem to prevent the attack just described. 12.8 A common management requirement is that "all external Web traffic must flow via the organization's Web proxy."

However, that requirement is easier stated than implemented. Discuss the various problems and issues, possible solutions, and limitations with supporting this requirement. In particular consider issues such as identifying exactly what constitutes "Web traffic" and how it may be monitored, given the large range of ports and various protocols used by Web browsers and servers. 12.9 Consider the threat of "theft/breach of proprietary or confidential information held in key data files on the system." One method by which such a breach might occur is the accidental/deliberate e-mailing of information to a user outside of the organization. A possible countermeasure to this is to require all external e-mail to be given a sensitivity tag (classification if you like) in its subject and for external e-mail to have the lowest sensitivity tag. Discuss how this measure could be implemented in a firewall and what components and architecture would be needed to do this. 12.10 You are given the following "informal firewall policy" details to be implemented using a firewall like that in Figure 12.2: 1. E-mail may be sent using SMTP in both directions through the firewall, but it must be relayed via the DMZ mail gateway that provides header sanitization and content filtering. External e-mail must be destined for the DMZ mail server. 2. Users inside may retrieve their e-mail from the DMZ mail gateway, using either POP3 or POP3S, and authenticate themselves. 3. Users outside may retrieve their e-mail from the DMZ mail gateway, but only if they use the secure POP3 protocol, and authenticate themselves. 4. Web requests (both insecure and secure) are allowed from any internal user out through the firewall but must be relayed via the DMZ Web proxy, which provides content filtering (noting this is not possible for secure requests), and users must authenticate with the proxy for logging. 5. Web requests (both insecure and secure) are allowed from anywhere on the Internet to the DMZ Web server. 6. DNS lookup requests by internal users allowed via the DMZ DNS server, which queries to the Internet. 7. External DNS requests are provided by the DMZ DNS server. 8. Management and update of information on the DMZ servers is allowed using secure shell connections from relevant authorized internal users (may have different sets of users on each system as appropriate). 9. SNMP management requests are permitted from the internal management hosts to the firewalls, with the firewalls also allowed to send management traps (i.e., notification of some event occurring) to the management hosts. Design suitable packet filter rulesets (similar to those shown in Table 12.1) to be implemented on the "External Firewall" and the "Internal Firewall" to satisfy the aforementioned policy requirements. 432 Appendix A Some Aspects of Number Theory A.1 Prime and Relatively Prime Numbers Divisors Prime Numbers Relatively Prime Numbers A.2 Modular Arithmetic A.1 / Prime and Relatively Prime Numbers 433 In this appendix, we provide some background on two concepts referenced in this book: prime numbers and modular arithmetic. A.1 Prime and Relatively Prime Numbers In this section, unless otherwise noted, we deal only with nonnegative integers. The use of negative integers would introduce no essential differences. Divisors We say that b ≠ 0 divides a if a = mb for some m, where a, b, and m are integers. That is, b divides a if there is no remainder on division. The notation b- a is commonly used to mean b divides a. Also, if b- a, we say that b is a divisor of a. For example, the positive divisors of 24 are 1, 2, 3, 4, 6, 8, 12, and 24. The following relations hold: ■■ If a- 1, then a = {1 ■■ If a- b and b- a, then a = {b ■■ Any b ≠ 0 divides 0 ■■ If b- g and b- h, then b-(mg + nh) for arbitrary integers m and n To see this last point, note that If b- g, then g is of the form g = b * g1 for some integer g1. If b- h, then h is of the form h = b * h1 for some integer h1. So mg + nh = mbg1 + nbh1 = b * (mg1 + nh1) and therefore b divides mg + nh. Prime Numbers An integer p 7 1 is a prime number if its only divisors are {1 and {p. Prime numbers play a critical role in number theory and in the techniques discussed in

Chapter 3. Any integer a 7 1 can be factored in a unique way as a = p1 a1 * p2 a2 * c * pt at where p1 6 p2 6 c 6 pt are prime numbers and where each ai is a positive integer. For example, 91 = 7 * 13 and 11011 = 7 * 112 * 13. It is useful to cast this another way. If P is the set of all prime numbers, then any positive integer can be written uniquely in the following form: a = p q∈P pap where each ap Ú 0 The right-hand side is the product over all possible prime numbers p; for any particular value of a, most of the exponents ap will be 0. 434 Appendix A / Some Aspects of Number Theory The value of any given positive integer can be specified by simply listing all the nonzero exponents in the foregoing formulation. Thus, the integer 12 is represented by {a2 = 2, a3 = 1}, and the integer 18 is represented by {a2 = 1, a3 = 2}. Multiplication of two numbers is equivalent to adding the corresponding exponents: k = mn S kp = mp + np for all p What does it mean, in terms of these prime factors, to say that a- b? Any integer of the form pk can be divided only by an integer that is of a lesser or equal power of the same prime number, pj with j ... k. Thus, we can say a- b S ap ... bp for all p Relatively Prime Numbers We will use the notation gcd(a, b) to mean the greatest common divisor of a and b. The positive integer c is said to be the greatest common divisor of a and b if 1. c is a divisor of a and of b. 2. Any divisor of a and b is a divisor of c. An equivalent definition is the following: gcd(a, b) = max[k, such that k- a and k- b] Because we require that the greatest common divisor be positive, gcd(a, b) = gcd(a, -b) = gcd(-a, b) = gcd(-a, -b). In general, gcd(a, b) = gcd(- a- , - b-). For example, gcd(60, 24) = gcd(60, -24) = 12. Also, because all nonzero integers divide 0, we have gcd(a, 0) = - a- . It is easy to determine the greatest common divisor of two positive integers if we express each integer as the product of primes. For example, 300 = 22 * 31 * 52 18 = 21 * 32 gcd(18, 300) = 21 * 31 * 50 = 6 In general, k = gcd(a, b) S kp = min(ap, bp) for all p Determining the prime factors of a large number is no easy task, so the preceding relationship does not directly lead to a way of calculating the greatest common divisor. The integers a and b are relatively prime if they have no prime factors in common, that is, if their only common factor is 1. This is equivalent to saying that a and b are relatively prime if gcd(a, b) = 1. For example, 8 and 15 are relatively prime because the divisors of 8 are 1, 2, 4, and 8, and the divisors of 15 are 1, 3, 5, and 15, so 1 is the only number on both lists. A.2 / Modular Arithmetic 435 A.2 Modular Arithmetic Given any positive integer n and any nonnegative integer a, if we divide a by n, we get an integer quotient q and an integer remainder r that obey the following relationship: a = qn + r 0 ... r 6 n; q = :a/n; where :x; is the largest integer less than or equal to x. Figure A.1 a demonstrates that, given a and positive n, it is always possible to find q and r that satisfy the preceding relationship. Represent the integers on the number line; a will fall somewhere on that line (positive a is shown, a similar demonstration can be made for negative a). Starting at 0, proceed to n, 2n, up to qn such that qn ... a and (q + 1)n 7 a. The distance from qn to a is r, and we have found the unique values of q and r. The remainder r is often referred to as a residue. If a is an integer and n is a positive integer, we define a mod n to be the remainder when a is divided by n. Thus, for any integer a, we can always write: a = :a/n; * n + (a mod n) Two integers a and b are said to be congruent modulon, if (a mod n) = (b mod n). This is written a K b mod n. For example, 73 K 4 mod 23 and 21 K -9 mod 10. Note that if a K 0 mod n, then n- a. The modulo operator has the following properties: 1. a K b mod n if n-(a - b) 2. (a mod n) = (b mod n) implies a K b mod n 0 n 2n 3n qn a (q + 1)n n (a) General relationship r 0 15 15 10 30 = 2 15 70 (b) Example: 70 = (4 15) + 10 45 = 3 15 60 = 4 15 75 = 5 15 Figure A.1 The Relationship a = qn + r; 0 ... r 6 n 436 Appendix A / Some Aspects of Number Theory 3. a K b mod n implies b K a mod n. 4. a K b mod n and b K c mod n imply a K c mod n. To demonstrate the first point, if n-(a - b), then (a - b) = kn for some k. So we can write a = b + kn. Therefore, (a mod n) = (remainder when b +

kn is divided by n) = (remainder when b is divided by n) = (b mod n). The remaining points are as easily proved. The (mod n) operator maps all integers into the set of integers {0, 1, c , (n - 1)}. This suggests the question: Can we perform arithmetic operations within the confines of this set? It turns out that we can; the technique is known as modular arithmetic. Modular arithmetic exhibits the following properties: 1. [(a mod n) + (b mod n)] mod n = (a + b) mod n 2. [(a mod n) - (b mod n)] mod n = (a - b) mod n 3. [(a mod n) * (b mod n)] mod n = (a * b) mod n We demonstrate the first property. Define (a mod n) = ra and (b mod n) = rb. Then we can write a = ra + jn for some integer j and b = rb + kn for some integer k. Then (a + b) mod n = (ra + jn + rb + kn) mod n = (ra + rb + (k + j)n) mod n = (ra + rb) mod n = [(a mod n) + (b mod n)] mod n The remaining properties are as easily proved. 437 Appendix B Projects for Teaching Network Security B.1 Research Projects B.2 Hacking Project B.3 Programming Projects B.4 Laboratory Exercises B.5 Practical Security Assessments B.6 Firewall Projects B.7 Case Studies B.8 Writing Assignments B.9 Reading/Report Assignments 438 Appendix B / Projects for Teaching Network Security Many instructors believe that research or implementation projects are crucial to the clear understanding of network security. Without projects, it may be difficult for students to grasp some of the basic concepts and interactions among components. Projects reinforce the concepts introduced in the book, give the student a greater appreciation of how a cryptographic algorithm or protocol works, and can motivate students and give them confidence that they are capable of not only understanding but implementing the details of a security capability. In this text, I have tried to present the concepts of network security as clearly as possible and have provided numerous homework problems to reinforce those concepts. However, many instructors will wish to supplement this material with projects. This appendix provides some guidance in that regard and describes support material available in the Instructor's Resource Center (IRC) for this book, accessible to instructors from Pearson Education. The support material covers nine types of projects: 1. Research projects 2. Hacking project 3. Programming projects 4. Laboratory exercises 5. Practical security assessments 6. Firewall projects 7. Case studies 8. Writing assignments 9. Reading/report assignments B.1 Research Projects An effective way of reinforcing basic concepts from the course and for teaching students research skills is to assign a research project. Such a project could involve a literature search as well as an Internet search of vendor products, research lab activities, and standardization efforts. Projects could be assigned to teams or, for smaller projects, to individuals. In any case, it is best to require some sort of project proposal early in the term, giving the instructor time to evaluate the proposal for appropriate topic and appropriate level of effort. Student handouts for research projects should include the following: ■■ A format for the proposal ■■ A format for the final report ■■ A schedule with intermediate and final deadlines ■■ A list of possible project topics The students can select one of the topics listed in the instructor's manual or devise their own comparable project. The IRC includes a suggested format for the proposal and final report as well as a list of fifteen possible research topics. B.3 / Programming Projects 439 B.2 Hacking Project The aim of this project is to hack into a corporation's network through a series of steps. The Corporation is named Extreme In Security Corporation. As the name indicates, the corporation has some security holes in it, and a clever hacker is able to access critical information by hacking into its network. The IRC includes what is needed to set up the Web site. The student's goal is to capture the secret information about the price on the quote the corporation is placing next week to obtain a contract for a governmental project. The student should start at the Web site and find his or her way into the network. At each step, if the student succeeds, there are