1

Network Security Overview

> *If you know your enemies and know yourself, you will win hundred times in hundred battles. If you know yourself but not your enemies, you will suffer a defeat for every victory won. If you do not know yourself or your enemies, you will always lose.*
>
> —Sun Tzu, "The Art of War"

The goal of network security is to give people the freedom to enjoy computer networks without the fear of compromising their rights and interests. Network security therefore needs to guard networked computer systems and protect electronic data that is either stored in networked computers or transmitted in the networks. The Internet, which is built on the IP communication protocols, has become the dominant computer network technology. It interconnects millions of computers and edge networks into one immense network system. The Internet is a public network, where individuals or organizations can easily become subscribers of the Internet service by connecting their own computers and networking devices (e.g., routers and sniffers) to the Internet and paying a small subscription fee.

Because IP is a store-forward switching technology, where data is transmitted using routers controlled by other people, user A can read user B's data that goes through user A's network equipment. Likewise, user A's data transmitted in the Internet may also be read by user B. Hence, any individual or any organization may become an attacker, a target, or both. Even if one does not want to attack other people, it is still possible that one's networked computers may be compromised into becoming an attacking tool. Therefore, to achieve the goal of network security, one must first understand the attackers, what could become their targets, and how these targets might be attacked.

1.1 Mission and Definitions

The tasks of network security are to provide *confidentiality*, *integrity*, *nonrepudiation*, and *availability* of useful *data* that are transmitted in public networks or stored in networked computers.

Introduction to Network Security: Theory and Practice, Second Edition. Jie Wang and Zachary A. Kissel.

The concept of data has a broad sense in the context of network security. Any object that can be processed or executed by computers is data. Thus, source code, executable code, files in various formats, email messages, digital music, digital graphics, and digital video are each considered data. Data should be read, written, or modified only by legitimate users. That is, unauthorized individuals or organizations are not allowed to have access to data.

Just as CPU, RAM, hard disk, and network bandwidth are resources, data is also a resource. Data is sometimes referred to as *information* or *messages*.

Each piece of data has two possible states, namely, the *transmission state* and the *storage state*. Data in the transmission state is simply data in the process of being delivered to a network destination. Data in the storage state is that which is stored in a local computer or in a storage device. Thus, the meanings of data confidentiality and data integrity have the following two aspects:

1. Provide and maintain the confidentiality and integrity of data that is in the transmission state. In this sense, confidentiality means that data during transmission cannot be read by any unauthorized user, and integrity means that data during transmission cannot be modified or fabricated by any unauthorized user.
2. Provide and maintain the confidentiality and integrity of data that is in the storage state. Within this state, confidentiality means that data stored in a local device cannot be read by any unauthorized user through a network, and integrity means that data stored in a local device cannot be modified or fabricated by any unauthorized user through a network.

Data nonrepudiation means that a person who owns the data has no way to convince other people that he or she does not own it.

Data availability means that attackers cannot block legitimate users from using available resources and services of a networked computer. For example, a computer system infected with a virus should be able to detect and disinfect the virus without much delay, and a server hit by denial of service attacks should still be able to provide services to its users.

Unintentional components in protocol specifications, protocol implementations, or other types of software that are exploitable by attackers are often referred to as *loopholes*, *flaws*, or *defects*. They might be an imperfect minor step in a protocol design, an unforeseen side effect of a certain instruction in a program, or a misconfigured setting in a system.

Defense is the guiding principle of network security, but it is a passive defense because before being attacked, the victim has no idea who the attackers are and from which computers in the jungle of the Internet the attackers will launch their attacks. After a victim is attacked, even if the attacker's identity and computer system are known, the victim still cannot launch a direct assault at the attacker, for such actions may be unlawful. What constitutes legal actions against attackers involves a discussion of relevant laws, which is beyond the scope of this book. Therefore, although offense may be the best defense in military operations, this tactic may not apply to network security. Building a deep layered defense system is instead the best possible defense tactic in network security. Within this type of defense system, multiple layers of defense mechanisms are used to resist possible attacks.

Network security is a major part of information security. In addition to network security, information security deals with many other security issues, including security policies, security auditing, security assessment, trusted operating systems, database security, secure code, emergency response, computer forensics, software forensics, disaster recovery, and security training.

- Security policies are special rules to protect a computer network system against security attacks. For example, security policies may specify what types of data are to be protected, who should be given the access right of read from or write to the data, and how the data should flow from one place to the next.
- Security auditing is a procedure of checking how well the security policies for a particular computer network system are followed. It may be a manual procedure or an automated procedure run by software tools.
- Security assessment is a procedure of determining the security needs of a particular system, measuring the strength and weakness of the existing security policies, and assessing whether the security policies are reasonable and whether security loopholes exist.
- A trusted operating system is an operating system without any security flaws or loopholes in system designs, computing resource management, software implementations, and configurations.
- Database security is a set of security measures specifically devised for database systems, specifying which data fields are accessible by which level of users.
- Secure software is software that contains no security flaws, loopholes, or side effects.
- Intrusion response is a set of actions that should take place when a computer network system is detected being intruded by intruders.
- Cyber forensics studies how to collect information of user activities from computer systems and network communications, providing evidence to indict cyber criminals. Cyber forensics can be further divided into computer forensics and network forensics.
- Disaster recovery is a set of mechanisms to bring a computer system that goes down because of attacks or natural disasters back to a working status.

This book does not cover these issues, but it may touch certain aspects of them.

1.2 Common Attacks and Defense Mechanisms

Common network security attacks can be characterized into a few basic types. Almost every known network security attack is either one of these basic types or a combination of several basic types.

1.2.1 Eavesdropping

Eavesdropping is an old and effective method for stealing private information. In network communications, the eavesdroppers may intercept data from network traffic using a networking device and a packet sniffer. A packet sniffer, or network sniffer, is a program for monitoring incoming network traffic. When connecting a router to the Internet, for example, one can use a packet sniffer to capture all the IP packets going through that router. `TCPdump` and `Wireshark` (formerly known as `Ethereal`) are network sniffers widely used today, which are available as free downloads (see Exercise 1.5).

Using a packet sniffer as an eavesdropping tool, one can intercept IP packets that go through the router he controls. To capture a particular IP packet, however, the eavesdropper must first determine which communication path the IP packet will travel through. Then, he could either try to get control of a certain router on the path or try to insert a new router of his own on the path. This task is more difficult but is not impossible. For example, the eavesdropper may try to compromise a router on the path and install a packet sniffer in it to intercept the IP packets he is after. The eavesdropper may also use an ARP spoofing technique (see Section 1.2.4) to reroute IP packets to his sniffer without compromising a router.

Eavesdropping wireless communications is easier. In this case, the attacker simply needs to place a receiver with the same radio frequency of the wireless network within the communication range of the network.

There is no way to stop eavesdropping in public networks. To counter eavesdropping, the best defense mechanism is to encrypt data. Computer cryptography is developed for this purpose, where the sender encrypts data into an unintelligible form before he transmits it. Data encryption is a major component of computer cryptography. It uses an encryption key in concert with an encryption algorithm, to break the original data into pieces and mix them up in a certain way to make it unintelligible, so that the eavesdropper cannot obtain any useful information out of it. Thus, even if the eavesdropper is able to intercept the encrypted data, he is still not able to obtain the original data without knowing the decryption key. We often refer the original data as *plaintext* data, or simply plaintext, and encrypted data as *ciphertext* data, or simply ciphertext.

Ciphertext data can be converted back to plaintext data using a decryption key along with a decryption algorithm. The encryption key is a string of characters, which is also referred to as *secret key*. In a symmetric-key encryption algorithm, also referred to as conventional encryption, the encryption key and the decryption key are identical. In a public-key encryption algorithm, also known as asymmetric-key encryption, the encryption key and the decryption key are different.

1.2.2 Cryptanalysis

Cryptanalysis is the art and science of finding useful information from ciphertext data without knowing the decryption keys. For example, in a substitution cipher that substitutes plaintext letters with ciphertext letters, if a ciphertext message reveals a certain statistical structure, then one may be able to decipher it. To obtain a statistical structure of the data, one may calculate the frequency of each character in the ciphertext data and compare it against the known statistical frequency of each character in the language used in the plain text. For example, in the English language, the letter “e” has the highest frequency. Thus, in a substitution cipher, the character that has the highest frequency in the ciphertext data is likely to correspond to the plaintext letter “e” (see e.g., Exercise 1.7). This analysis can be further extended to common phrases. Analyzing statistical structures of ciphertext messages was an effective method to break encryptions before the computer era.

Modern encryption algorithms can produce ciphertext without any trace of statistical structure. Therefore, modern cryptanalysis is focused on analyzing encryption algorithms using mathematical techniques and high-performance computers.

The best method against cryptanalysis is to devise encryption algorithms that reveal no statistical structures in ciphertext messages using sophisticated mathematics and longer encryption keys. Using sophisticated mathematics makes mathematical analysis difficult. Using longer keys makes brute force attacks impractical. In addition to having stronger encryption algorithms, it is equally important to distribute and manage keys safely and to implement encryption algorithms without exploitable loopholes.

1.2.3 Password Pilfering

Computer users need to prove to the system that they are legitimate users. The most widely used authentication mechanism is in the form of user names and user passwords. User names are public information, but user passwords must be kept secret. Only two parties should have knowledge of the password, namely, the user and the underlying computer program (e.g., an operating system or a specific software application). A password is a sequence of letters, digits, or other characters, which is often selected by the user. Legitimate users enter their user names and passwords to prove their legitimacy to the computer program. An unauthorized user may impersonate a legitimate user to "legitimately" log on to a password-protected system or application, if he can get hold of a legitimate user name and password pair. He can then gain all the "legal" rights to transmit, receive, modify, and fabricate data.

Password protection is often the first defense line, and sometimes, it may be the only defense mechanism available in the system. Thus, we must take measures to ensure that user passwords are well protected against larcenies. For this purpose, we will look at several common methods for pilfering user passwords. These methods include *guessing*, *social engineering*, *dictionary attacks*, *side-channel attacks*, and *password sniffing*. *Phishing* attacks and *pharming* attacks have become the most common form of mass social engineering attacks in recent years.

1.2.3.1 Guessing

Guessing is the simplest method to acquire a password illegitimately. The attacker may get lucky if users use short passwords or if they forget to change the default passwords created for them. Also, users have a tendency to use the same passwords.

According to data compiled yearly by SplashData, a password management company, the top 10 most common passwords used by users, listed in decreasing order of popularity, are as follows:

1. `123456`
2. `password`
3. `12345678`
4. `qwerty`
5. `abc123`
6. `123456789`
7. `111111`
8. `1234567`
9. `iloveyou`
10. `adobe123`

If the user chooses a simple password such as these 10 easy ones, then the guesser would indeed have an easy task.

1.2.3.2 Social Engineering

Social engineering is a method of using social skills to pilfer secret information from the victims. For example, attackers may try to impersonate people with authority or organizations of reputation to trick unvigilant users to reveal their user names and user passwords to the attackers. Impersonation may be carried out either in person or in an electronic form. Phishing and pharming are common electronic forms of social engineering attacks in recent years, targeted at a large number of people.

There are other forms of social engineering attacks. For example, attackers may try to collect recycled papers from the recycle bins in a corporation's office building, hoping to find useful login information. Attackers may also make a Web browser pop up a window asking for user login information.

Physical Impersonation

Physical impersonation means that the attacker pretends to be a different person to delude the victim. For example, the following imaginary conversion between the attacker and a receptionist named Betty demonstrates how a social engineering attack might be carried out in person:

> Attacker: (Speaking with an authoritative voice.) "Hello, Betty, this is Nina Hatcher. I am Marketing Manager of the China branch office."
>
> Betty: (Thinking that this woman knew my name, my number, and spoke like a manager, she must be whom she said she was.) "Hello, Nina, what can I do for you?"
>
> Attacker: "Betty, I am attending a meeting in Guangzhou to finalize an important deal with a large corporation in China. To close the deal, I'll need to verify certain technical data produced by your group that I believe is still stored in the computer at your site. This is urgent. I tried to log on to your system today, but for some reason it didn't work. I was able to log on to it yesterday though. Is your computer down? Can you help me out here?"
>
> Betty: "Well, I don't know what happened. But you may try the following $\cdots$" (Thinking that she is doing the company a favor by telling the marketing manager how to get into the system.)

Phishing

Phishing attacks are mass social engineering attacks that take advantage of people with a tendency to trust authorities. The main forms of phishing attacks are disguised email messages or masqueraded Websites. For example, attackers (also called *phishers*) send disguised email messages to people as if these messages were from banks, credit card companies, or other financial institutions that people may pay attention to. People who receive such messages are told that there was a security breach in their accounts, and so they are required to verify their account information for security purposes. They are then directed to a masqueraded Website to enter their user names and passwords (e.g., see Exercise 1.15). The following example is a real phishing message verbatim (The reader may notice a number of grammatical errors and format problems.):

> From: UML NEW EMAIL <helpdesk@uml.edu>
>
> To:
>
> Date: Wed, Jul 7, 2010 at 2:28 AM
>
> Subject: Re UNIVERSITY I.T.S UPDATE
>
> Welcome to the university of Massachusetts Lowell New webmail system.
>
> Many of you have given us suggestions about how to make the Umass Lowell webmail better and we have listened. This is our continuing effort to provide you with the best email services and prevent the rate of spam messages received in your inbox folder daily. Consequently all in-active old email accounts will be deleted during the upgrade.
>
> To prevent your account from deletion and or being suspended we recommends all email accounts owner users to upgrade to the new email. Fill in your data in the blank space provided;
>
> (Email:_______), (User I.D_______), (password_______)
> (Retype password____________).
>
> The University I.T.S
>
> www.uml.edu
>
> Checked by AVG - Version: 8.5.437 Virus Database: 271.1.12840 - Release

This was a blunt phishing attack, in which the phisher simply asked the recipients to fill in the blanks with their passwords. Other more sophisticated phishing emails may contain a bogus Website as a trap to capture account information entered by the victims. Here, the email and the Website are the baits. The sniffing mechanisms hiding behind the Web page are the hook. Most phishing emails, no matter how well they are put together, would often contain the lines of “Something happened with your account, and you need to go to this page to fix it, or your account will be deleted”. In general, any phishing email would contain a link to a bogus Website, called a *phishing site*. Phishing sites may look like the real ones, with the purpose of luring careless users to enter useful login information only to be captured by the phisher.

Even if you do not plan to enter any information on the bogus Website, clicking the link in the phishing email may already compromise your computer, for modern phishing techniques make it possible to embed exploits in a Web page, and the exploits will be activated when you open the Web page.

Users may look at the following three things to detect abnormalities: (1) the “From” address, which may look odd; (2) the URL links the phishers want them to click on, which may be similar to but definitely different from the real site (e.g., a URL that looks like Citicard is in reality not the Citibank’s real site); and (3) the look and feel of the Website if the user fails to identify any abnormality during the first two items, for the bogus Website would not be exactly the same as the real site. For example, the color scheme may look different. If you receive an email from a bank or a credit card company telling you that your have a problem with your account and asking you for your user name and password, then most likely it is a phishing email, for banks or credit card companies would never send emails to their customers asking for their account information.

Sometimes, a phishing email may contain a line similar to this: “To be removed from this list click here.” Do not click on this link, for it will notify the attacker that the user did read the email and consequently more annoying emails may come.

Antiphishing extensions of Web browsers are emerging technology for detecting and blocking phishing sites. *Email scanners* may also be used to identify phishing emails. However, blocking phishing and not blocking legitimate emails is challenging, even with appropriate email scanners. Thus, users may also want to develop their own tools to detect compromised email accounts and disable them before they can send out phishing emails.

1.2.3.3 Pharming

Pharming attacks use Web technologies to redirect users from the URLs they want to visit to a URL specified by the attacker, including changing DNS setting or the hosts file on the victim’s computer, where DNS stands for domain-name service. Attacks that change DNS settings are also referred to as DNS poisoning. If an DNS-poisoning attack is launched from an insecure home router or wireless access point, it is also referred to as a drive-by pharming. Reported by Symantec in 2008, the first drive-by pharming attack was targeted at a Mexican bank.

Similarly to phishing attacks, pharming may also be used to pilfer user passwords. But pharming attacks do not need to set up baiting messages as phishing attacks normally do and hence may disguise themselves better and trap people in more easily.

To counter pharming attacks, it is important for users to make sure that their DNS software and the hosts files have not been compromised and that the URL they are visiting is the right one before doing anything else.

1.2.3.4 Dictionary Attacks

For security reasons, only encrypted passwords, that is, not in their original form, should be stored in a computer system. This prevents attackers from learning the passwords even if they break into the system. In early versions of UNIX and Linux operating systems, for example, the encrypted user passwords of the system are stored in a file named `passwd` under directory `/etc`. This encryption is not a one-to-one encryption. Namely, the encryption algorithm can calculate the ciphertext string of a given password, but the ciphertext string cannot be uniquely decrypted. Such an encryption is also referred to as an *encrypted hash*. In early versions of UNIX and Linux operating systems, user names and the corresponding encrypted user passwords stored in the `passwd` file were ASCII strings that could be read by users. In later versions of UNIX and Linux operating systems, however, the encrypted user passwords of the system are no longer stored this way. Instead, they are stored in a file named `shadow` under directory `/etc`, which is an access-restricted system file.

In the Windows NT/XP operating system, for another example, the user names and the encrypted user passwords are stored in the system’s registry in a file named `SAM`. They can be read using special tools, for example, `pwdump`.

Dictionary attacks take advantage of the way some people use dictionary words, names, and dates as passwords. These attacks find user passwords from their encrypted forms. A typical dictionary attack proceeds as follows:

1. Obtain information of user names and the corresponding encrypted passwords. This was done, for example, in early versions of Unix or Linux by getting a copy of the

`/etc/passwd` file. In Windows XP, it can be done using `pwdump` to read the system registry.

2. Run the encryption routine used by the underlying system on all dictionary words, names, and dates. That is, compute the encrypted hash for each dictionary word, each name, and each date.
3. Compare each output obtained from Step 2 with the encrypted passwords obtained from Step 1. If a match presents, a user password is found. In other words, suppose that w is a word and $w' = crypt(w)$ is the output of the encryption routine *crypt* on input w. Suppose that u and p_u are a pair of user name and encrypted password of user u. If $w' = p_u$, then w is user u's password or is equivalent to user u's password, for w may not be unique.

Step 2 is computationally intensive, for there are many words, names, and dates. To avoid carrying out this costly computation each time an encrypted hash is given, one would want to precompute Step 2 and store the results (i.e., password-hash pairs) in one table, so that one only needs to do a table lookup to find the corresponding plaintext password from the given encrypted hash. But such a table will be humongous. Constructing a *Rainbow table* helps to reduce the table size and make the computation at Step 2 manageable.

Rainbow Tables

A rainbow table is a table of two columns constructed as follows: let r be a function that maps an encrypted hash of a password to a string in the domain of possible passwords. This function r is referred to as a *reduction* function, for the length of a password is typically shorter than the length of its encrypted hash value. The function r can be defined in a number of ways. For example, suppose that the domain of passwords is a set of all possible eight-character strings. Let h be a cryptographic hash function that, on an eight-character password, generates a 16-character long hash value. Then, we may define r as follows: For any eight-character string w, function r on input $h(w)$ returns the last eight characters of $h(w)$. Function r may also return the first eight characters of $h(w)$ or any combination of eight characters selected from $h(w)$. Note that r is not an inverse function of h.

Let w_{11} be a given password. Apply h and r alternatively to obtain a chain of passwords that are different pairwise:

$$w_{11}, w_{12}, \cdots, w_{1n_1},$$

where n_1 is a number chosen by the user, and

$$w_{1i} = r(h(w_{1,i-1})),$$
$$i = 2, 3, \cdots, n_1.$$

Store

$$(w_{11}, h(w_{1n_1}))$$

in the rainbow table, where w_{11} is in the first column and $h(w_{1n_1})$ is in the second column. Figure 1.1 depicts the construction of a rainbow table.

Now, choose a new password w_{21} (i.e., w_{21} has not been generated in previous chains). Repeat the same procedure for another round to obtain

$$w_{22}, w_{23}, \cdots, w_{2n_2},$$

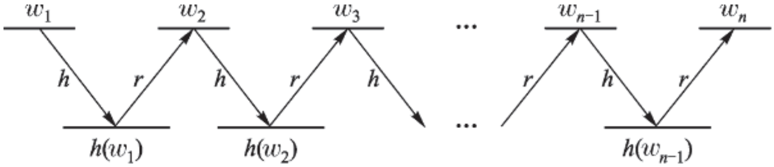

Figure 1.1 Construction of a rainbow table

where n_2 is a number chosen by the user and $w_{2i} = r(h(w_{2,i-1})$ for $i = 2, 3, \cdots, n_2$, such that the first chain and the second chain are disjoint. That is, for any $1 \le u \le n_1$ and $1 \le v \le n_2$, we have $w_{1u} \ne w_{2v}$. Store

$$(w_{21}, h(w_{2n_2}))$$

in the rainbow table. Performing this procedure k times will generate k rows in the rainbow table as follows:

Password	Hash value
w_{11}	$h(w_{1n_1})$
w_{21}	$h(w_{2n_2})$
$\cdots$	$\cdots$
w_{k1}	$h(w_{kn_k})$

where w_{j1} is the first password in the jth chain, $h(w_{jn_j})$ is the encrypted hash of the last password in the same chain, and the chains are disjoint pairwise.

Let $f : A \to B$ and $g : B \to A$ be two functions. Let $y \in B$ and $i \ge 0$. Define $(f \circ g)^i(y)$ as follows:

$$(f \circ g)^i(y) = \begin{cases} y, & \text{if } i = 0, \\ f(g((f \circ g)^{i-1}(y))), & \text{if } i \ge 1. \end{cases}$$

Let Q_0 be an encrypted value of a password w. That is, $Q_0 = h(w)$. If

$$h((h \circ r)^i(Q_0)) = h(w_{jn_j})$$

for some $i \ge 0$ and some j with $1 \le j \le k$ and $i \le j$, then w is possible to appear in the jth chain of $w_{j1}, \cdots, w_{jn_j}$. Thus, the following algorithm may help find w.

1. Set $Q_1 \leftarrow Q_0$ and $t \leftarrow 0$. Let $n = \max\{n_1, \cdots, n_k\}$.
2. Check if there is a $1 \le j \le k$ such that $Q_1 = h(w_{jn_j})$ and $t \le n$. If yes, goto Step 3; otherwise, goto Step 4.
3. Apply r and h alternatively on w_{j1} for $0 \le i \le j$ times until $w_{jn_i} = (r \circ h)^i(w_{j1})$ is generated such that $h(w_{jn_i}) = Q_0$. If such a w_{jn_i} is found, return $w = w_{jn_i}$; otherwise, goto Step 4.
4. Set $Q_1 \leftarrow h(r(Q_1))$ and $t \leftarrow t + 1$. If $t \le n$, then goto Step 2. Otherwise, return "password not found." (The rainbow table does not contain the password whose hash value equals Q_0.)

Note that we may use several different reduction functions in the same password chain, which helps avoid collisions that two different chains, starting from different passwords, may end up at the same password or at the same hash value at some point.

Remarks

It is worth noting that dictionary attacks may also be used in a positive way. For example, Windows Office allows users to encrypt Microsoft Word documents, where secret keys used for encryption are generated on the basis of the passwords selected by users. If, after a long while, a user forgets the password of a password-protected document, then the file will no longer be useful, for the user cannot decrypt it. To solve this problem, a company named Elcomsoft developed a password recovery software program using the dictionary attack techniques. This is a positive application of dictionary attacks. On the other hand, we note that if an encrypted office document is stolen, then the thief can also use this program to decrypt the document. There is a positive side and a negative side to every thing. A kitchen knife is intended to chop food, but it can also be used to harm people. Water can carry boats, but it can also topple them.

We also note that the file `/etc/passwd` in recent versions of UNIX and Linux no longer displays the encrypted user passwords (see Exercise 1.8). This makes it more difficult for the attackers to obtain the list of encrypted passwords for launching a dictionary attack.

1.2.3.5 Password Sniffing

Password sniffers are software programs used to capture remote login information such as user names and user passwords. Common network applications such as Telnet, FTP, SMTP, and POP3 often require users to type in their user names and passwords for authentication, making it possible for a password sniffer to intercept useful login information. For remote logins, however, one may use special programs (e.g., SSH) to encrypt all messages, thus making it more difficult to sniff user passwords.

SSH and other programs that encrypt login information such as HTTPS, however, are still vulnerable to password sniffing attacks. For example, Cain and Abel, a password recovery tool for the Microsoft Operating Systems, is a network sniffing tool that can capture and crack encrypted passwords using dictionary, brute-force, and cryptanalysis attacks. Cain & Abel can be downloaded free of charge from `http://www.oxid.it/cain.html`.

1.2.3.6 Side-Channel Attacks

Social media sites, such as Facebook, LinkedIn, and Twitter, provide user-friendly platforms for billions of users to interact with each other. Many users also like to post their personal data on social media sites for others to see. However, security measures on social media sites are not as strong as one would like. As a result, it is often easier to obtain user login information from social media sites than from online banking sites. In June 2012, for example, LinkedIn was under a massive attack from Russia, resulting in 6 million user passwords stolen, for the passwords were not encrypted properly.

In general, attackers can legitimately obtain personal information posted by users from social media sites, including favorite food, pets, siblings, birthdays, and birthplaces, as well as the schools they graduated from, and the places they grew up in. Many of these items are the typical questions the users are asked to verify their identity when logging to their banking accounts. To make things worse, people tend to use the same passwords for multiple accounts, including their banking accounts. Thus, social media has become a side channel for attackers to obtain user passwords of relevant banking accounts.

1.2.3.7 Key-Logging Attacks

A Key logger is software that records key strokes of the user at the point of entry. Eavesdropping keystrokes is a more effective method to capture passwords entered by the user on the keyboard before the passwords are encrypted. Pressing a key on the keyboard will also generate radiation, which may be exploited to learn keystrokes. Attacks such as this are referred to as tempest attacks. We may use anti-key-logging software tools to counter key-logging attacks.

1.2.3.8 Password Protection

The following rules and practices can help protect passwords from pilfering:

1. Use long passwords, with a combination of letters, capital letters, digits, and other characters such as $, #, &, %. Do not use dictionary words, common names, and dates as passwords. This rule makes guessing attacks and dictionary attacks arduous.
2. Do not reveal your passwords to anyone you do not know. Do not submit to anyone who acts as if he has authority. If you have to give out your password to someone you trust, do so face to face. Avoid telling passwords over the phone or using email. This practice helps prevent social engineering breaches.
3. Change passwords periodically and do not reuse old passwords. This rule helps defend users against patient and persistent attackers who may keep on running dictionary attacks on all possible strings formed using the first rule and hope that they may get lucky. Attackers may also keep records of old passwords they have identified.
4. Do not use the same password for different accounts. Thus, even if a user's password for a particular account is compromised, the user's other accounts would still be safe.
5. Do not use remote login software that does not encrypt user passwords and other important personal information. This practice makes password sniffing difficult.
6. Shred all discarded papers using a good paper shredder. This practice makes it difficult for attackers to find useful information from discarded old documents.
7. Avoid entering any information in any popup window, and avoid clicking on links in suspicious emails. Instead, go to the legitimate Website directly using the true URL address, and follow the directions there. This practice helps counter password sniffing and reduce the chance of being caught by phishers.

1.2.3.9 Other User-Authentication Methods

Authentication using user passwords is so far the most widely used authentication method.

Traditionally, there are three methods for proving one's identity. The first method uses secret passwords. The second method uses biometrics of unique biological features, for example, fingerprints and retinas. The third method uses authenticating items, for example, passes and certificates of identification. These three methods have been applied and implemented in computer applications.

The first method is implemented in the form of user names and user passwords.

The second method is implemented in the form of connecting biometric devices to a computer, for example, fingerprint readers and retina scanners. These devices are relatively more expensive to acquire and maintain and so are often used in a tightly controlled environment

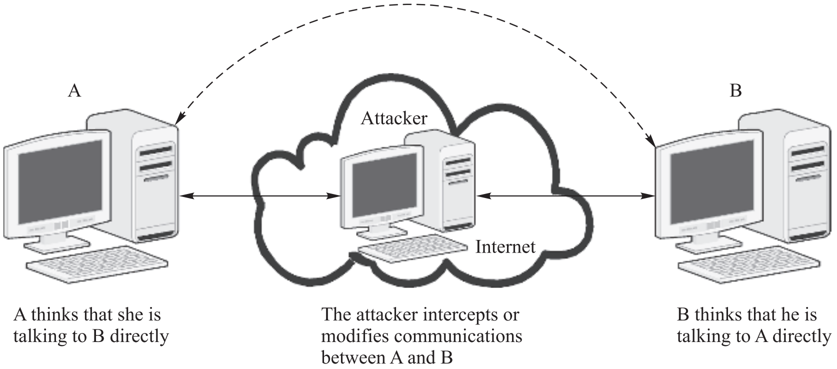
A
B
Attacker
Internet
A thinks that she is talking to B directly
The attacker intercepts or modifies communications between A and B
B thinks that he is talking to A directly

Figure 1.2 Man-in-the-middle attacks. The solid lines represent the actual communications, and the dash line represents the perceived communication between user A and user B

where high levels of security are required. For example, instead of using credit card readers at check-out stands to authenticate credit holders and link payments to their accounts, using fingerprint readers is just as convenient and is more secure.

The third method is implemented in the form of electronic passes authenticated by the issuer. Certain authentication protocols (e.g., Kerberos) use this method to authenticate users.

Authentication using user passwords is the easiest method to implement and so far the most commonly used authentication method.

1.2.4 Identity Spoofing

Identity spoofing attacks allow attackers to impersonate a victim without using the victim's passwords. Common identity spoofing attacks include *man-in-the-middle* attacks, *message replays*, *network spoofing*, and *software exploitation* attacks.

1.2.4.1 Man-in-the-middle Attacks

In a man-in-the-middle attack, the attacker tries to compromise a network device (or installs one of his own) between two or more users. Using this device, the attacker can intercept, modify, or fabricate data transmitted between users. The attacker will then forward them as if they have not been touched by the attacker. For example, the attacker may intercept an IP packet sent by user A, modify its payload, and then send the modified packet to user B as if it comes from user A. This way, both users may still believe that they are directly talking to each other, without realizing that the confidentiality and integrity of the IP packets they receive have already been compromised (see Fig. 1.2).

Encrypting and authenticating IP packets are common measures to thwart man-in-the-middle attacks. This is because the attacker cannot read or modify an encrypted IP packet without decrypting it. Also, the attacker has no way to authenticate a modified or fabricated IP packet to convince the receiver that it comes from a legitimate sender.

1.2.4.2 Message Replays

In a message replay attack, the attacker first intercepts a legitimate message, keeps it intact, and then retransmits it at a later time to the original receiver. In some authentication protocols, for example, after user A proves herself to the system as a legitimate user, she will be given an authentication pass. With this pass, she will be able to obtain services provided by the system. This pass is encrypted, and so it cannot be modified. However, the attacker may intercept it, keep a copy, and use it later to impersonate user A to get the services from the system.

The following are common mechanisms for thwarting message replay attacks:

1. Attach a random number to the message. This number is referred to as a *nonce*. When a user receives a message whose nonce appeared before, he knows that this message is a replay, which is then discarded. This method, however, requires that users keep a record of every nonce they first encounter, which may not be practical.
2. Attach a time stamp to the message. When a user receives a message whose time stamp is old, he knows that this message is a replay. This method, however, requires that all networked computers be synchronized with little error. While not a problem in local area networks, accurate synchronization is difficult to achieve in wide area networks.
3. The best method to thwart message replay attacks is to use a nonce and a time stamp together. Using this method, synchronization does not have to be very accurate, and the user only needs to keep track of the nonces he encounters in a short and fixed time interval. The user stores a nonce in a record with a time stamp when it is first recorded. When this time stamp becomes old, the nonce is removed. The length of the time interval is determined by the worst-case error of an achievable synchronization. A message is considered as a replay only when its nonce is already in the record or its time stamp is out of the time interval.

1.2.4.3 Network Spoofing

IP spoofing is one of the major network spoofing techniques. It consists of *SYN flooding*, *TCP hijacking*, and *ARP spoofing*. ARP spoofing is also referred to as *ARP poisoning*.

SYN Flooding

SYN flooding exploits an implementation side effect of the TCP/IP network protocols. In a SYN flooding attack, the attacker fills the target computer's TCP buffer with a large volume of SYN control packets, making the target computer unable to establish communications with other computers. When this happens, the target computer is called a *muted* computer or a *silenced* computer. The TCP buffer is a set of contiguous memory locations allocated by the underlying network application program. It is used to store TCP packets that have been received but not yet processed.

To launch a SYN flooding attack against a target computer, the attacker sends to it a large number of crafted SYN packets, each requesting to establish TCP connections. The term *crafted SYN packet* means that the source address contained in the SYN packet is a legitimate IP address, but the host computer on that address is not reachable. This host computer may be powered off or taken off the network. We call such a computer a *dead* computer. Detecting whether an IP address is unreachable can be done using the `ping` command (or other commands in case a live computer has been hardened to not respond to the `ping` command). If an IP address does not respond to `ping`, then it is probably unreachable. The `ping` command is a common network management tool based on the ICMP protocol. The attacker uses crafted SYN packets to avoid being tracked down. And he uses a legitimate source IP address to ensure that the crafted SYN packets will be delivered to its destination, because the domain name server will drop IP packets with fake IP addresses.

According to the three-way handshake procedure in the TCP protocol, the victim's computer is obliged to send an ACK packet to the source IP address contained in the SYN packet it receives and waits for an ACK packet to be sent back from that IP address. However, the host computer with that source IP address is not reachable, and so it will not respond. Thus, the victim's computer will never receive the ACK packet it is waiting for, forcing the crafted SYN packet to remain in the TCP buffer until its lifetime expires. During this period of time, the TCP buffer is completely occupied by (i.e., flooded with) crafted SYN packets, and so the victim's computer will have no room in the TCP buffer to establish any new connection with another computer. The victim's computer is then considered muted.

TCP Hijacking

Suppose that computer V is a company computer and user A is an employee of that company and is going to log on to computer V from home. User A's computer sends a SYN control packet to V and now suppose that an attacker intercepts this packet. The attacker then uses the SYN flooding attack to mute computer V, so that V cannot complete the three-way handshake protocol with user A's computer. If the attacker can predict the correct TCP sequence number for the ACK packet that is supposed to be sent to A from the muted computer V, then the attacker can craft an ACK packet and send it to user A's computer. The crafted ACK packet uses the correct TCP sequence number and V's IP address as the source IP address. User A's computer receives the ACK packet and verifies that it has the correct TCP sequence number. It then sends an ACK packet to the attacker to complete the three-way handshake procedure with the attacker. Thus, the TCP connection that user A's computer has established is with the attacker, instead of with V.

To see how this works, we note that the TCP protocol uses the sequence number in its TCP header to identify which TCP packets belong to the same communication. Figure 1.3 depicts the TCPv4 header format. As the TCP protocol header does not contain the source IP address, the TCP-layer software would not check the legitimacy of the IP addresses contained in the IP header. See Fig. 1.4 for the standard IPv4 header format. The IP protocol routes

<table>
<tr><td colspan="3">16-bit source port number</td><td>16-bit destination port number</td></tr>
<tr><td colspan="4">32-bit sequence number</td></tr>
<tr><td colspan="4">32-bit acknowledgement number</td></tr>
<tr><td>4-bit hdr length</td><td>6 reserved bits</td><td>6 control bits</td><td>16-bit window size</td></tr>
<tr><td colspan="3">16-bit TCP checksum</td><td>16-bit urgent pointer</td></tr>
</table>

Figure 1.3 The standard TCPv4 header format

<table>
<tr><td>4-bit version</td><td>4-bit hd length</td><td>8-bit type of service (TOS)</td><td colspan="2">16-bit total length (in bytes)</td></tr>
<tr><td colspan="3">16-bit identification number</td><td>3-bit flags</td><td>13-bit fragmentation offset</td></tr>
<tr><td colspan="2">8-bit time to live (TTL)</td><td>8-bit protocol</td><td colspan="2">16-bit header checksum</td></tr>
<tr><td colspan="5">32-bit source IP address</td></tr>
<tr><td colspan="5">32-bit destination IP address</td></tr>
</table>

Figure 1.4 The standard IPv4 header format

the IP packet it receives to the destination on the basis of the information contained in the IP header. It does not keep track of the header information of previous IP packets it received. Thus, checking the source IP address at the IP layer does not help identify whether the source IP address in the current IP packet is the same as those in previous IP packets. This shows that the working of the TCP/IP protocol suite (its early implementation in particular) actually makes TCP hijacking possible. To stop TCP hijacking, it is important to use software (e.g., `TCP wrappers`) that checks IP addresses at the TCP layer.

In 1994, Kevin Mitnick, a resident in North Carolina of the United States, launched TCP hijacking attacks from his home and broke into several major companies' computers a few thousand kilometers away in California. Mitnick was later convicted and sentenced to 5 years in prison for this crime.

ARP Spoofing

Computers are identified by unique media access control (MAC) addresses. MAC addresses are also called physical addresses. ARP is an address resolution protocol at the link layer, which converts the destination IP address in the IP header to the MAC address of the underlying computer at the destination network. In an ARP spoofing attack, the attacker changes the legitimate MAC address of an IP address to a different MAC address chosen by the attacker (see, e.g., Exercise 1.7.2).

To prevent ARP spoofing attacks, checking is the key. In particular, we should strengthen checking procedures of MAC addresses and domain names and make sure that the source IP address and the destination address in an IP packet have not been changed during transmissions.

1.2.5 Buffer-Overflow Exploitations

Buffer overflow, also referred to as *buffer overrun*, is a common software loophole exploited by attackers. A buffer is a set of contiguous memory locations allocated to a process. The size of the buffer is fixed in its declaration in the program. A buffer overflow occurs if the process writes more data into the buffer than it can hold. The following is a simple C program that writes the `buffer` of eight bytes with a string `str` of 34 bytes, causing it to overflow.

```
int main() {
  char buffer[8];
  char *str = "This is a test of buffer overflow.";
  strcpy(buffer, str);
  printf("%s", buffer);
}
```

It is possible to exploit buffer overflows to redirect the victim's program to execute attackers' own code located in a different buffer area. Such attacks often exploit function calls in standard memory layout, where the buffer is placed in a *heap* and the return address of the function call is placed in a *stack*. The stack is in the higher end of the memory space, while the heap is in the lower end, where they grow toward each other and shrink away from each other (see Fig. 1.5). The following are general steps of this type of attacks:

1. Find a program that is vulnerable to buffer overflows. For example, programs that use string-based functions (e.g., `strcpy()` and `strcat()`) are vulnerable, for they do not check bounds. These functions would copy as many characters as possible until a NULL byte is encountered.
2. Figure out the address of the attacker's code.
3. Determine the number of bytes that is long enough to overwrite the return address.
4. Overflow the buffer that rewrites the original return address of the function call with the address of the attacker's code.

In reality, exploiting buffer overflows to breach security is often a complex and difficult procedure.

The best way to prevent buffer overflow attacks is to close the doors of overflow. That is, one should always add statements to check bounds when dealing with buffers in a program. Avoid using string functions that do not check bounds.

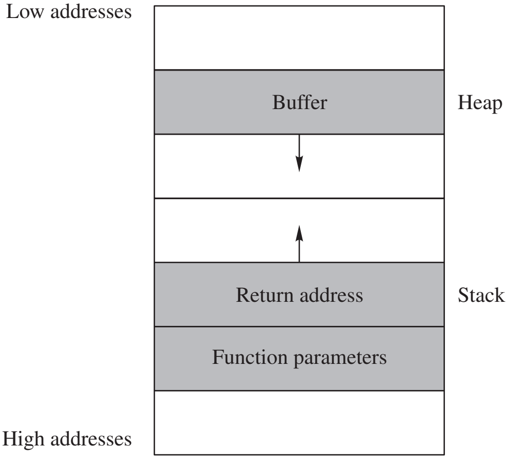

Figure 1.5 Typical memory layout for function call

1.2.5.1 Compiler Protections for Buffer Overflow

Buffer overflow often has a simple programming fix. Unfortunately, these fixes are often overlooked. To combat this problem, compiler-induced protections have been developed, one of which is the notion of *canary values*. Borrowing its name from the coal-mining practice of lowering a canary into a coal mine to determine if sufficient oxygen exists for miners to enter, a canary value is a special value stored on the programs execution stack, which helps to detect if the return address from a function has been altered. This value is pushed on to the stack immediately after the return address. If a buffer-overflow attack is executed, the heap will likely be overflown into the canary value *and* the return address (see Fig. 1.6). Thus, if the attacker manages to overwrite the return address, then it is likely that they will also overwrite the canary value and thus be detected.

To enable the canary value to protect from buffer overflow, the function prologue and epilogue code generated by the compiler must be modified to deal with the canary value. The prologue must be modified such that the canary value is pushed onto the stack after the return address. The epilogue code must be modified to check that the canary value is valid.

If the same canary value is used in every program, every time a function call is made, the attacker would easily be able to construct buffer overflow attacks. To launch an attack on the system that uses the same canary value for every function call, the attacker merely places the canary value in the correct location in the data used for buffer overflow. The check of the canary in the function epilogue will pass, and thus the return address will vector off to the attacker's malicious code. To correct this problem, a random canary value is often used. A random canary value is chosen at execution and used for just that execution. This means that every time the attacker runs the potentially vulnerable code, the canary is different, and thus the attacker cannot use the attack that works with the fixed canary values.

1.2.6 Repudiation

In some situations, the owner of the data may not want to admit ownership of the data to evade legal consequences. He may argue that he has never sent or received the data in question.

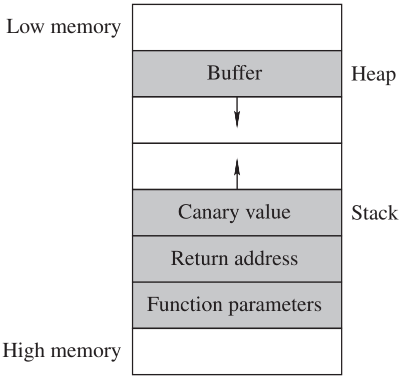

Figure 1.6 Typical memory layout for a function call that uses a canary value

Repudiation is straightforward if the data has not been authenticated. Even if the data has been authenticated, repudiation is still possible when the underlying authentication methods or the communication protocols contain loopholes. The owner of the authenticated data may be able to convince the judge that, because of the loopholes, anyone could have easily fabricated the message and made it look like it was produced by him.

Secure encryption and authentication algorithms are effective mechanisms to counter repudiation attacks.

1.2.7 Intrusion

Intrusion in network security means that an illegitimate user, also known as intruder, gains access to someone else's computer systems. The intruder may turn a victim's computer into his own server, which may result in stolen computing resources and network bandwidth from the victim. The intruder may also steal useful information residing in the victim's computer.

Configuration loopholes, protocol flaws, and software side effects may all be exploited by intruders. Opening TCP or UDP ports that should not be open is a common configuration loophole. TCP and UDP ports are entry points of network application programs.

Intrusion detection is a technology for detecting intrusion incidents. Closing TCP and UDP ports that may be exploited by intruders can also help reduce intrusions.

1.2.7.1 IP Scans and Port Scans

IP scans and *port scans* are common hacking tools. IP scans search for existing IP addresses in the Internet, and port scans search for open ports in a computer. Attackers use IP scans to search for potential targets and port scans to identify open ports that are vulnerable in the targets.

However, IP scans and port scans can also help users to identify in their own systems which ports are open and which ports may be vulnerable. Several such products are available. For example, `ShieldsUP!` of Gibson Research Corporation and `Nessus` of Southwest Research Institute are two such products (see Exercise 1.19).

1.2.8 Traffic Analysis

The purpose of *traffic analysis* is to determine who is talking to whom by analyzing IP packets. Even if the payload of the IP packet is encrypted, the attacker may still obtain useful information from analyzing IP headers. An IP header contains the source IP address and the destination IP address, which reveal who is sending messages to whom. If its payload (i.e., the encapsulated TCP packet) is not encrypted, the port numbers can also be obtained. This information can be used to learn which application program is used to read the message. When preparing for a big event, individuals or organizations may frequently exchange messages before the event takes place. If the traffic analyzer learns this information from analyzing IP headers, an attacker may conclude that something big is about to happen.

The best way to combat traffic analysis is to encrypt IP headers. But an IP packet with an encrypted IP header cannot be routed to the destination. Thus, a new plaintext IP header must be inserted in front of the encrypted IP header for delivery. This may be done using a network *gateway*. A gateway is a special-purpose computer shared by many users in the local network. It

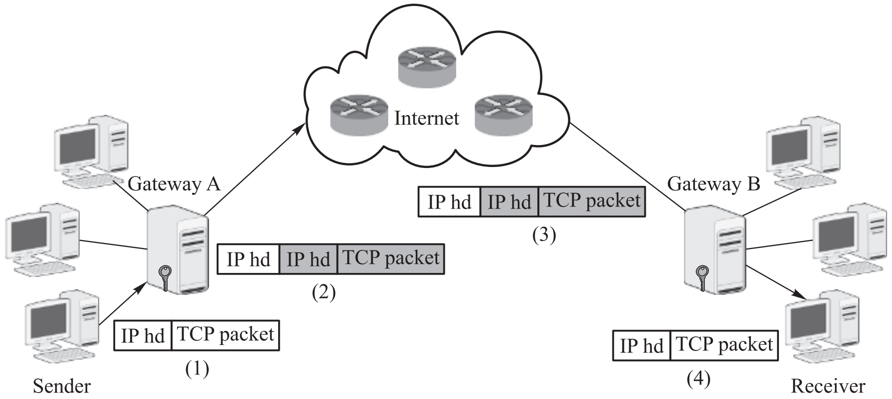

Figure 1.7 Using gateways to encrypt IP packets. (1) Sender forwards an IP packet to gateway A at the sending side. (2) Gateway A encrypts sender's IP packet (the shaded part) and routes it to the next router in the Internet. (3) The IP packet from Gateway A is delivered to gateway B at the receiving side, with certain attributes (e.g., TTL) in the plaintext IP header (shown as the unshaded part) modified. (4) Gateway B removes its header, decrypts the encrypted IP packet of the sender, and forwards it to the receiver

can encrypt a user's IP packet (including its header) at the sending side, decrypt the encrypted IP packet at the receiving side, and forward it to the destination MAC address. If there are no other routers between the sending-side gateway and the sender's computer, and there are no other routers between the receiving-side gateway and the receiver's computer, then traffic analysis can only reveal that the two gateways are talking to each other (see Fig. 1.7), without gaining any information about which user behind one gateway is talking to which user behind the other gateway.

1.2.9 Denial of Service Attacks

The goal of *denial of service attacks* is to block legitimate users from getting services they can normally get from servers. Such attacks often force the target computer to process a large number of useless things, hoping to consume all its critical resources. A denial of service attack, denoted by DoS, may be launched from a single computer, or from a group of computers distributed in the Internet. The latter attack is called a *distributed denial of service attack* and is denoted by DDoS.

1.2.9.1 DoS Attacks

SYN flooding is a typical and effective technique used by DoS attacks. The `smurf` attack is another typical type of DoS attack, where `smurf` is the name of the software used to execute the attack. It sends an excessive number of messages to the target computer and crashes it by consuming all its resources. In a typical `smurf` attack, the attacker sends crafted `ping`

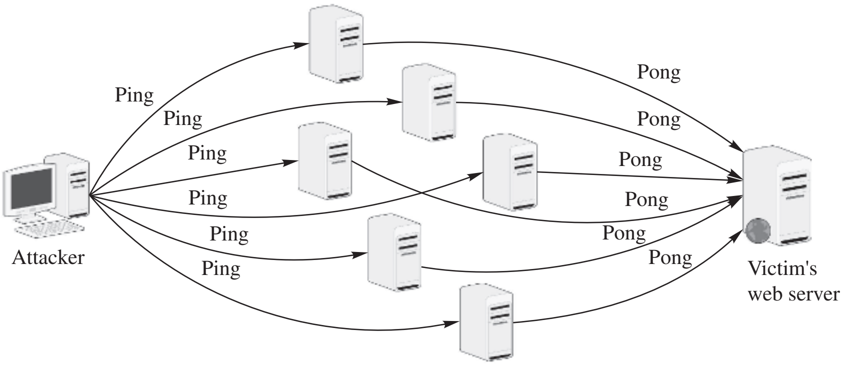

Figure 1.8 Smurf attack

requests to a large number of computers within a short period of time, where the source IP address in the crafted `ping` request is replaced with the victim's IP address. According to the ICMP protocol, a computer that receives a `ping` request will respond to the source IP address with a `pong` message, informing the sender that "I am alive". Therefore, each computer that receives the crafted `ping` request will respond to the victim's computer with a `pong` message. Forced to process a large number of `pong` messages within a short period of time, the victim's computer will use up its computing resources and crash (see Fig. 1.8). Thus, the idea of `smurf` attacks is to crash a single target with a lot of borrowed hammers.

1.2.9.2 DDoS Attacks

A typical DDoS attack proceeds according to the following sequence:

1. Compromise as many networked computers as possible. This may be achieved using Trojans (see Section 1.2.10 for a description of Trojans).
2. Install special software in the compromised computers to carry out a DoS attack at a certain time later. Such software is called *zombie software*, and such a computer is called a *zombie computer* or simply a *zombie*. A collection of zombies is also called a *zombie army*, which is now typically called a *botnet*.
3. Issue an attack command to every zombie computer to launch a DoS attack on the same target at the same time.

Figure 1.9 depicts a DDoS attack. On receiving the attacker's command, each zombie computer uses SYN flooding to mute the victim's Website.

In 2000, for example, a 15-year-old high-school student in Montreal, Canada, with an assumed name "Mafiaboy," launched a DDoS attack against Web servers of several major companies and paralyzed these Web servers for a week. These companies, including Amazon, Cable News Network, eBay, E*Trade, Dell, and Yahoo!, suffered substantial financial losses because of this attack. Mafiaboy was sentenced to spend 8 months in a youth detention center.

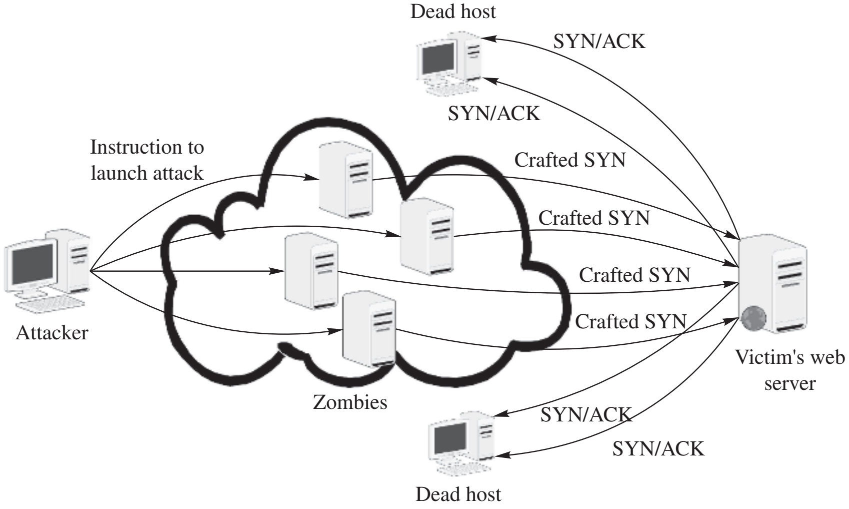

Figure 1.9 A DDoS attack using SYN flooding to mute the victim's Website

1.2.9.3 Spam Mail

Spam mails are uninvited emails, which may be commercial messages or phishing messages. While not intended to bring the victim's computer out of service, spam mails do consume computing resources. Spam mails are annoying, particularly when one's mailbox is filled up with them.

Standard electronic messaging systems have made it possible for individuals and companies to send unwanted bulk messages to people. Such individuals or companies are often referred to as *spammers*. Spamming can occur in any form of network applications, but email spam is by far the most common spamming form. According to a recent statistics, about half a billion spam emails are sent in every single day. In other words, each email user is expected to receive about eight spam messages a day. Spamming also occurs in Web search engines, Instant Messaging, blogs, mobile phone messaging, and other network applications.

Spam filters are software solutions to detect and block spam mails from reaching the user's mailbox.

1.2.10 Malicious Software

Software intended to harm computers is *malicious software*. Malicious software is also referred to as *malware*. Common forms of malicious software include *virus*, *worms*, *Trojans*, *logic bombs*, *backdoors*, and *spyware*.

1.2.10.1 Viruses and Worms

A computer virus is a piece of software that can reproduce itself. However, a virus is not a standalone program. In other words, it must attach itself to another program or another file.

A program or file that contains a virus is called an *infected program* (also called an *infected host*). When an infected program is transmitted to another computer, the virus that lives in it is also transmitted along with its host program.

The execution of a virus is initiated by the infected host. Namely, only when an infected program is executed or an infected file is opened, a virus contained in it may get executed. When executed, a virus may do harm (e.g., delete system files) to the system where its host resides or replicate itself to infect other healthy hosts in the system.

A computer worm is also a piece of software that can reproduce itself. Unlike a virus, a worm is a standalone program. In other words, it does not need a host to live in. A worm can execute itself at any time it wishes. When executed, a worm may do harm to the system where it resides or replicate itself to other systems through networks.

There are two common measures to combat viruses and worms. One measure deploys *virus scans* to detect, quarantine, and delete infected hosts and worms. The other measure, consisting of the following rules, blocks viruses and worms from entering a computer:

1. Do not download software (e.g., games) from untrusted Websites or other sources.
2. Do not open any executable file given to you by someone you do not know.
3. Make sure that software patches are installed and up to date.

   Neglecting software patches may be fatal. For example, in the summer of 2001, many systems that run Microsoft Internet Information Services (IIS) were hit by the Code Red worm, the Nimda worm, and the Code Red II worm. These worms made headline news, and they all exploited the same loophole in IIS. Microsoft knew about this problem and provided a patch to correct it a year earlier. However, many system administrators did not install this patch and thus left wide open doors into their systems for the worms to come in and do damage.

1.2.10.2 Trojans

Trojans are also called Trojan horses. The name Trojan horse came from a Greek legend. Legend has it that ancient Greeks, wanting to apprehend a beauty named Helen, attacked the fortified city of Troy but failed. Faking a retreat, the Greeks left behind a huge, hollow wooden horse with a number of soldiers hidden inside. Not suspecting any danger, the Trojans hauled the wooden horse inside the city as a trophy. At night, the Greek army returned, and the soldiers hidden inside the wooden horse went out and opened the city gates for the invasion troops to come in. The city of Troy fell.

In the realm of network security, Trojans are software programs that appear to do one thing but secretly also perform other tasks. Trojans often disguise themselves as desirable and harmless software applications to lure people to download them. When they are executed by the user, the hidden functions contained in them, which now have the user's access rights, do harmful things secretly. Games and network management tools available for free downloads from unknown Websites often are Trojans. Trojans may also use appealing names such as `AntiSPYware.exe` or `Real_Player.exe` (note that the real one is `RealPlayer.exe`) to trap users to use them.

The same measures of combating viruses and worms can also be used to combat Trojans. Virus scans can also detect, quarantine, and delete Trojans.

1.2.10.3 Logic Bombs

Logic bombs are subroutines or instructions embedded in a program. Their execution is triggered by conditional statements. For example, a company employee working on a development project may install a logic bomb inside a program. The bomb will be set off only if the employee has not run the program in a certain period of time. When that condition is met, it would mean that the employee was fired some time before. The logic bomb in this case is used to gain revenge against the employer.

There are three measures to counter logic bombs. First, employers should always do their best to take care of their employees, so that none would be tempted to place a logic bomb. Second, project managers should hire an outside company or form a special team of reviewers from a different group of people other than the developers to review the source code. Third, relevant laws should be established so that employees who planted logic bombs will face criminal charges. With these countermeasures in place, unhappy employees would think twice before planting logic bombs in programs.

1.2.10.4 Backdoors

Backdoors are secret entrance points to a program. They are often inserted by software developers to provide a short cut to enter a password-protected program when attempting to modify or debug code. These backdoors that avoid the typical password entrances of normal users may later be discovered and used by attackers. Attackers who compromise network systems have been known to insert their own backdoors so that they can more easily re-enter later.

We note that, with the increase of outsourcing software development projects and other vital tasks to other countries, the potential for logic bombs and backdoors also increases. The major counter measure of backdoors is to check source code, which should be conducted by an independent team.

1.2.10.5 Spyware

Spyware is a type of software that installs itself on the user's computer. Spyware is often used to monitor what users do and to harass them with popup commercial messages. *Browser hijacking* and *zombieware* are the most disastrous kinds of spyware.

Browser Hijacking

Browser hijacking is a technique that changes the settings of the user's browsers. It may replace the user's default Website with a different Website selected by the attacker. Or it may stop the user from visiting the Websites he or she wants to visit. For example, the Google redirect virus, which affected a lot of people in 2012/2013, redirects the browser to a Website that has nothing to do with the search query entered by the user.

Zombieware

Zombieware is software that takes over the user's computer and turns it into a zombie for launching DDoS attacks or into a relay that carries out harmful activities such as sending spam email or spreading viruses. Therefore, the purpose of zombieware is to hijack computers.

In addition to hijacking browsers and computers, spyware can also do a number of other things, including the followings:

Monitoring

Spyware can be used to monitor and report to a Web server or to the attacker's machine a user's surfing habits and patterns, such as which Web pages the user has browsed and which products the user has purchased.

Password Sniffing

Spyware can be used to sniff user passwords by logging users' keystrokes using a keystroke logger. A keystroke logger is a program that can capture user names and user passwords when the users type them in.

Adware

Adware is software that automatically displays advertising materials on the user's computer screen. The common form of adware is popup windows with commercial material. While not intended to harm users, adware consumes user's precious computing resources and is annoying.

To counter spyware, users may use antispyware software to detect and block spyware. Microsoft's Windows Defender, for example, is such a software tool. Windows Defender is available as a free download.

Most modern antivirus software includes checks for spyware, adware, and hacking tools such as keystroke loggers and network sniffers.

1.3 Attacker Profiles

Attackers are often characterized as *black-hat hackers*, *script kiddies*, *cyber spies*, *employees*, and *cyber terrorists*.

1.3.1 Hackers

Hackers are people with special knowledge of computer systems. They are interested in subtle details of software, algorithms, and system configurations. Hackers are an elite group of well-trained and highly motivated people. Depending on their motives, hackers are further characterized as *black-hat hackers*, *white-hat hackers*, and *grey-hat hackers*.

1.3.1.1 Black-Hat Hackers

Black-hat hackers are people who hack computing systems for their own benefit. For example, they may hack into an online store's computer system and steal credit card numbers stored in it. They may then use the stolen credit card numbers to buy merchandise or sell them to other people. Black-hat hackers are the wicked doers in network security.

Note that, without the "black-hat" modifier, hacker is not a derogatory term. News media, however, have widely used hackers to denote black-hat hackers. To avoid confusions, several authors have suggested to use *crackers* to denote black-hat hackers.

1.3.1.2 White-Hat Hackers

White-hat hackers are hackers who have high moral standards. They hack computing systems for the purpose of searching for security loopholes and developing solutions. They publish security problems and solutions at security conferences, on dedicated Websites, or through special mailing lists. White-hat hackers are the righteous doers in network security.

1.3.1.3 Grey-Hat Hackers

Grey-hat hackers are hackers who wear a white hat most of the time but may also wear a black hat once in a while. For example, when they discover attacks, instead of reporting the incidents to law enforcements, grey-hat hackers may take the matter in their own hands and strike the attackers back themselves. Grey-hat hackers are the Robin Hood type people in the world of network security.

1.3.1.4 Disclosures of Security Problems

When discovering security vulnerabilities in a software product, white-hat hackers and grey-hat hackers would often work directly with the vendors of products to help them fix the problems before they release the details of their discoveries. Whether a full disclosure of their findings should be allowed is an ongoing debate, in part due to the perceived view of the white-hat hackers and the grey-hat hackers that the vendors are not doing enough to fix security problems in a timely manner.

1.3.2 Script Kiddies

Script kiddies are people who use scripts and programs developed by black-hat hackers to attack other people's computers. Such scripts and programs are often referred to as *hacking tools*. Script kiddie is a derogatory term. It is used to indicate that script kiddies only know how to copy and use a hacking tool. They do not understand how it works, and they are not capable of writing any hacking tool themselves. Script kiddies like to crack any target they possibly can, so that they can say to others in the underground cracker community that "I am smarter." Script kiddies may also attack targets with high profiles just to attract the attention of the media.

Although they do not know how to write hacking tools or understand how an existing hacking tool works, script kiddies are dangerous. Many of them are just teenagers who do not care about, or are not mature enough to know, the consequences of their actions. However, they are energetic, and they are everywhere. They launch attacks from unexpected places and at any time, which could inflict serious damages to other people.

1.3.3 Cyber Spies

Cyber espionage takes place at all levels. It could be an individual activity or an organizational effort. Cyber spies collect intelligence through intercepted network communications. They could be working for a good cause or just for money.

Governments run cyber intelligence units to intercept network communications and decipher encrypted messages. The National Security Agency (NSA) and the Central Intelligence Agency (CIA), for example, are the two largest intelligence agencies of the U.S. government. The NSA hires many first-class mathematicians and computer scientists to work for it. Many of them are professors at U.S. universities. They teach during school years and work for NSA during summers. They study encryption algorithms and develop cryptanalysis tools. This sort of work has helped win battles.

During World War II, for example, the intelligence department of the U.S. Pacific Fleet was able to partially decipher Japanese secret code, which helped Admiral Chester W. Nimitz, the Commander in Chief of the Pacific Fleet, deduce the Japanese scheme of invading the Midway Atoll in the mid-Pacific. Nimitz seized the opportunity and ordered his two aircraft carriers to ambush the approaching Japanese invasion forces. With another barely restored carrier joining in the battle a few days later, American aviators sunk four Japanese carriers, with the cost of losing only one carrier. The battle of Midway became a turning point, from a defensive to an offensive campaign for American Pacific naval forces.

1.3.4 Vicious Employees

Vicious employees are people who intentionally breach security to harm their employers. They may plant logic bombs or open backdoors in programs they help develop. They may act as script kiddies to attack company computers to get the attentions of their employers. They may also act as cyber spies to collect and sell company secrets for money.

1.3.5 Cyber Terrorists

Terrorists are extremists who do not hesitate to use extreme means to destroy public property and take innocent life. Cyber terrorists are terrorists who use computer and network technologies to carry out their attacks and produce public fear. Attacks by cyber terrorist have not been reported yet. However, if they did attack, cyber terrorists would be extremely harmful.

1.3.6 Hypothetical Attackers

The hypothetical attackers this book deals with are black-hat hackers, script kiddies, greedy cyber spies who are willing to betray their countries or organizations for monetary benefits, and vicious employees. Attackers of these four kinds may be wicked, but they are not terrorists. Cyber terrorists, on the other hand, are the die-hard enemies, and so they may need to be dealt with using a different set of measures not addressed in this book.

1.4 Basic Security Model

The basic security model consists of four components: *cryptosystems*, *firewalls*, anti-malicious-software *software* (AMS software), and intrusion detection systems (IDS system). Figure 1.10 shows this security model.

Cryptosystems use computer cryptography and security protocols to protect data. Security protocols include encryption protocols, authentication protocols, and key management

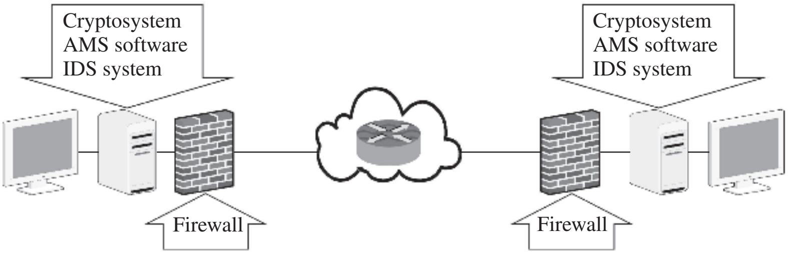

Figure 1.10 Basic security model

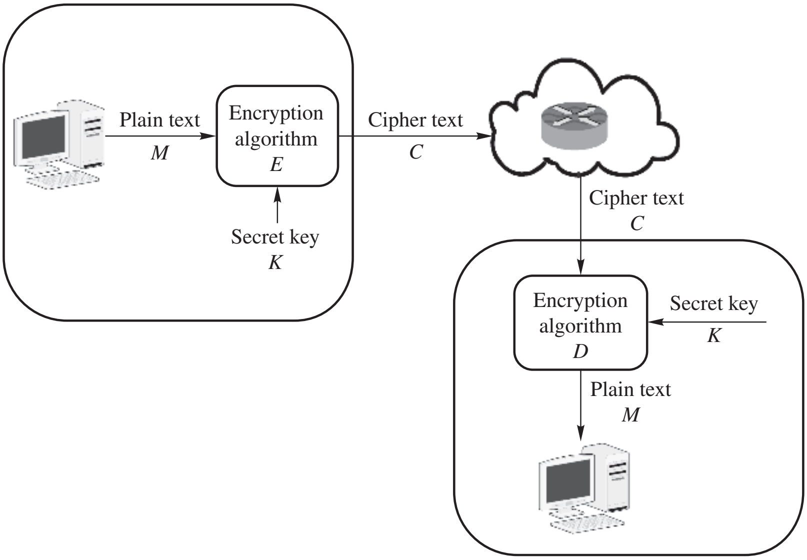

Figure 1.11 Network model of cryptosystem

protocols. Figure 1.11 shows the encryption and decryption components. It is customary to use E to denote an encryption algorithm, D its decryption algorithm, and K the secret key.

Firewalls, AMS software, and IDS systems are used to protect data stored in networked computers. Firewalls are special software packages installed in computers and networking devices that check incoming and outgoing network packets. Certain features of firewalls have also been incorporated into hardware devices to achieve faster processing speeds. AMS software scans system directories, files, and registries to identify, quarantine, or delete malicious code. IDS systems monitor system logins, study user behaviors, and analyze log files to identify and sound alarms when intrusions are detected.

In addition to using firewalls, AMS software, and IDS systems, we may also set up sacrificial decoy machines to lure attackers' attentions away from important computers. Decoy machines are also known as *honeypots*.

This book is centered around these four major components. This book also introduces honeypot technologies.

1.5 Security Resources

Network security is not something that can be taken care of once and for all, because when old security problems are solved, new security problems will appear. Thus, network security defenders will have to fight against the attackers continuously. Network security is an art of defense in digital form. This book covers basic principles, methods, and techniques of network security. It does not and cannot cover every aspect of the area. It does not and cannot tell you what the new attacks are going to be. Fortunately, there are many online security resources available to help you win this fight. The following are a few popular resources.

1.5.1 CERT

Founded in 1988, CERT is a research institute affiliated with Carnegie Mellon University. Its full name is Computer Emergency Response Team. Its budget comes mainly from the U.S. government.

CERT was the earliest organization devoted to studying security problems and offering practical solutions to system administrators to help secure their computer systems. It sends monthly reports to subscribers, free of charge, of any security breach identified in the current month, with recommended solutions. In addition, CERT also trains computer security personnel. Its Website is `www.cert.org`.

1.5.2 SANS Institute

Founded in 1989, SANS Institute is a nonprofit organization devoted to collecting, archiving, and publishing computer security information. It provides this information to users free of charge. SANS stands for SysAdmin, Audit, Network, and Security. In addition, SANS Institute also offers computer security training, issues certification, and funds research. Its Website is `www.sans.org`.

1.5.3 Microsoft Security

Microsoft security is Microsoft's official Website devoted to providing security information for Microsoft products. It provides security updates to Microsoft users. Its Website is `www.microsoft.com/security/default.mspx`.

1.5.4 NTBugtraq

NTBugtraq is a moderated open list service for users to post and discuss security exploits and bugs in Microsoft's products. Its Website is `www.ntbugtraq.com`.

1.5.5 Common Vulnerabilities and Exposures

The Common Vulnerabilities and Exposures (CVE) database is a free database maintained by the Mitre Corporation. CVE tracks software vulnerabilities across all major software products from all major vendors. This is the most widely used collection of information on security vulnerabilities. The vulnerabilities contained within the database are scored and ranked using the Common Vulnerability Scoring System (CVSS), a standard maintained by NIST. The CVE Web site is `www.cve.mitre.org`.

1.6 Closing Remarks

Sun Tzu said: All warfare is based on deception. Attackers may attack us where we are unprepared and appear where they are not expected. Network security is no exception. For example, even if we develop an unbreakable encryption algorithm, if keys are not managed properly, attackers can still break the encryption system, not by attacking the encryption algorithm, but by exploiting loopholes in key management protocols.

We must assume that attackers are capable of using any means available to achieve their objectives. They avoid what is strong and strike at what is weak. Therefore, we must remember that it will only take a small blow at a weak spot to bring down any apparently strong defense system. Also, a defense system would just be an ornament if one could bypass it. The famous Maginot Line, for instance, is an example. During World War II, the French militaries were confident that the Maginot Line of concrete fortifications they spent 10 years to build along the French-German border could stop German aggression. The German invasion forces, however, did not assault the Maginot Line directly as anticipated by the French. Instead, they dispatched motorized troops to quickly cut through the Low Countries of Belgium and the Netherlands and invaded France from unexpected locations in a third country. Lessons like this have taught us that in network security, we must constantly examine our network defense mechanisms from all aspects and fortify any weak point as soon as it is identified.

1.7 Exercises

1.7.1 Discussions

1.1. Have you experienced any network security attack described in the text? If so, please share your experience with the class. If you have experienced network security attacks not described in the text, please describe them in detail.

1.2. How did you solve the network security problems you encountered?

1.3. Why type of attackers do you think attacked you?

1.4. Networked computers are managed by different types of people. What type of people do you think are most vulnerable to network security attacks?

1.5. Why do you think phishing and pharming attacks are so common? What measures would your suggest to counter them?

1.6. Why do you think network security must be a multiple-layer defense mechanism?

1.7.2 Homework

1.1. This book assumes that the reader has taken a computer network course, or has sufficient experience working with computer networks.

(a) Describe the major structure of a TCP packet and explain the main functions of the TCP headers.

(b) Describe the major structure of an IP packet and explain the main functions of the IP headers.

(c) Explain the three-way handshake protocol in the TCP protocol and describe its main functions.

(d) Describe the difference between UDP and TCP. Give an example of an application that would use UDP and an application that would use TCP. Justify your answers.

1.2. On the basis of your understandings of network protocols, answer the following questions:

(a) Explain the main functions of the ARP protocol.

(b) Explain the main functions of the ICMP protocol.

(c) Explain the major functions of routers, switches, and gateways.

(d) Explain the major functions of the SMTP protocol.

1.3. Describe the major differences between IPv4 and IPv6.

1.4. Use network administration tools to familiarize yourself with network configurations.

(a) In the Windows operating system, `ipconfig`, `ping`, `tracert`, `nslookup`, and `netstat` are common network administration tools. On a machine running Windows, go to the `start` menu, select `run`, and then enter `cmd` to open a command window. Execute these five network administration tools. Explain the results you observe. For each of these admin tools, use option `-?` to list each option of the tool and explain its usage. For example, enter `ipconfig -?` to learn all options of `ipconfig` and explain their usage.

Execute the following commands and explain the results you observe:

```
ping cs.uml.edu
ping www.google.com
tracert www.yahoo.com
netstat -e
```

(b) In the UNIX and Linux operating systems, `ping`, `nslookup`, `netstat`, and `arp` are common network administration tools. You may use the `man` tool to find out how to use these tools. For example, enter `man netstat` to list all information about `netstat`. On a machine running UNIX or Linux, execute these tools and explain the results you observe.

(c) Open a `cmd` window on a Windows machine and execute `ipconfig /all` to list all information of the network setup of your PC. Write down the host name, MAC address of your network adapter, IP address, subnet mask, and default gateway of your PC.

In the UNIX and Linux operating systems, you may find the IP addresses of all hosts in the system in /etc/hosts. On a machine running UNIX or Linux, enter more /etc/hosts and explain what you see.

(d) Open a cmd window on a Windows machine and execute netstat -ano. Identify which ports are TCP ports, which ports are listening, which ports have established connections, and which ports are UDP ports. Also identify what programs are running on these ports.

To find out what program is running on a given port number, first identify its PID (process ID), and then open the Windows Task Manager window (e.g., you may open it by pressing the three keys of Ctrl-Alt-Del simultaneously). Select View, Select Columns,···, and PID. Then select Process and find out which program is running on the PID. For example, suppose that the following line is included in the result returned from netstat -ano:

```
Proto Local Address   Foreign Address State       PID
TCP   127.0.0.1:1026 127.0.0.1:1027  ESTABLISHED 664
```

From here, we know that Port 1026 is a TCP port where a connection has been established and its PID is 664. From the Windows Task Manager, we find out that postgres.exe has PID 664. Thus, we know that postgres.exe is running on Port 1026.

(e) Open a cmd window on a Windows machine and execute arp -a. It lists the physical address of your router. Compared to the physical address given by ipconfig /all, what is the difference between these two physical addresses? On a UNIX machine, enter arp -a on the UNIX prompt to list the ARP table in your machine.

1.5. Network sniffers are also referred to as packet sniffers. Network sniffers are software used to monitor network connections and obtain information of network packets. TCPdump and Wireshark are widely used packet sniffers with free downloads from www.tcpdump.org and www.wireshark.org, respectively. TCPdump has been around for many years. Wireshark, formerly known as Ethereal until 2006, is newer and has a nicer GUI interface.

If you are using a Windows machine, download from http://www.wireshark.org/ and install Wireshark-win64-1.12.0.exe (64-bit) or Wireshark-win32-1.12.0.exe (32-bit) or its newest version. This version contains WinPCap4.0.1. You will need to install it as well. If you are using other operating systems, please download and install from the Wireshark Website a corresponding version of Wireshark. Then execute Wireshark.

We want to sniff ARP packets. For this purpose, on the open window of "The Wireshark Network Analyzer," select Capture, Options, and then select network card in the Interface box. In the Capture Filter empty box type in arp, and then select Start to launch ARP sniffing. At this time, you will see a popup window titled "(the name of the network card): Capturing - Wireshark". To generate ARP packets (so that you have something to sniff), open a Web browser and visit a few Websites. After a short while, you will see that ARP packets have been captured in the popup window. Select Capture on the menu bar, then select

`Stop` to stop sniffing. Note that the Wireshark window is divided into three portions. The upper portion shows the ARP packets that have been captured, the middle portion shows the packer headers, and the lower portion shows the contents of the ARP packets in hexadecimal and ASCII code. Explain what you see.

Disclaimer: Network sniffing should only be done on a network where one has permission to do so and all parties are aware that it is (or may be) occurring. Otherwise, it may inadvertently break the Federal electronic eavesdropping and wiretap laws.

1.6. We often want to use a network sniffer to only pick up the types of packets we are interested in.

(a) Execute `Wireshark`. Select `Options` from the menu of `Capture`. A window named "`Wireshark: Capture Options`" will pop up. In the empty box of `Capture Filter`, enter `tcp port 25`, and then click `Start` to begin sniffing. Send yourself an email message. Then click `Capture` on the menu bar and select `Stop`. Explain what you see.

(b) Execute `Wireshark`. Select `Options` from the menu of `Capture`. A window named "`Wireshark: Capture Options`" will pop up. In the empty box of `Capture Filter`, enter `tcp port 80`, and then click `Start` to begin sniffing. Open a Web browser to visit a few Websites. Then select `Capture` on the menu bar and select `Stop`. Explain what you see.

1.7. Finding statistical structures in a cipher text message is a common cryptanalysis method. For example, given a ciphertext message, we first calculate the frequency of each letter occurring in the messages. We then compare these letter frequencies with the letter frequencies one would expect to have in the underlying language. If there is a clear one-to-one correspondence, we will then know which ciphertext letter corresponds to which plaintext letter. This method is especially effective to break earlier designed encryption algorithms.

In the English language, for example, the following table lists the expected frequency of each letter, in the decreasing order of frequencies.

e	t	a	o	i	n	s	h	r	d
12.702	9.056	8.167	7.507	6.996	6.749	6.327	6.094	5.987	4.253
l	c	u	m	w	f	g	y	p	b
4.052	2.782	2.758	2.406	2.360	2.228	2.015	1.974	1.929	1.492
v	k	j	x	q	z				
0.978	0.772	0.153	0.150	0.095	0.074				

If the ciphertext message is not long enough, we may not be able to obtain a frequency curve similar to that of the statistical frequency curve. Thus, we may also want to calculate frequencies of strings of two or more letters, for they may correspond to common letter strings such as er, or, the, and ing. Such information would be useful. Suppose that we have the following ciphertext message with punctuation and space removed, where the plain-text message is written in English:

```
NTCGPDOPANFLHJINTOOFITOVJHJCTMMHIHEMTCPFDWTSOFSHTOGFWTE
TTJJTBTOOFSZOVEOCHCVCHPJHOCGTOHNQMTOCNTCGPDCGFCSTQMFBTO
FBGFSFBCTSHJCGTQMFHJCTYCXHCGFAHYTDDHAATSTJCBGFSFBCTSHJC
GTBHQGTSCTYCCGHONTCGPDQSTOTSWTOCGTMTCCTSASTRVTJBZHJCGTQ
MFHJCTYCFJDOPPJTBFJOTFSBGAPSCGTQMFHJCTYCASPNFIHWTJBHQGT
SCTYCEZBPNQFSHJICGTASTRVTJBZPATFBGMTCCTSFIFHJOCCGTLJPXJ
BPNNPJASTRVTJBZHJCGTVJDTSMZHJIMFJIVFIT
```

(a) Calculate the frequency of each letter.

(b) Compare your calculated letter frequencies with the statistical letter frequencies, and find out the plaintext message properly punctuated and spaced.

1.8. In early versions of UNIX and Linux operating systems, login passwords of the users are stored in the file `/etc/passwd` in the following format:

```
user:password:ID:group-ID:comment:home:shell
```

where the encrypted passwords were readable text strings (e.g., $3/25\#2\%v$), making dictionary attacks possible. Recent versions have fixed this problem by only showing a symbol $*$ or x indicating that the user is required to enter the password. Suppose that your `/etc/passwd` file contains the following entry:

```
nobody:*:65534:10:NFS Nobody (normal):/:/bin/nosh
```

Explain the meaning of each component in this entry.

1.9. Let *h* be a hash function and *r* a reduction function. Let *T* be a rainbow table of *k* rows for *D* under *h* and *r*, where the *j*th row is $(w_{j1}, h(w_{jn_j})$ for $1 \leq j \leq k$. Let $Q_0 = h(w)$ and $Q_1 = (h \circ r)^i(Q_0)$, where $i \geq 0$. Suppose $Q_1 = h(w_{jn_j})$ for some $1 \leq j \leq k$ and $i \leq j$. Answer the following questions:

(a) Under what conditions will w appear in the *j*th chain of $w_{j1}, \cdots, w_{jn_j}$?

(b) Under what conditions will w not appear in the *j*th chain of $w_{j1}, \cdots, w_{jn_j}$?

(c) We note that in practice, *h* often maps a shorter password to a longer hash value. Thus, without lost of generality, we may assume that *h* is one-to-one for a given domain of passwords. It is common practice to use different reduction functions to produce a password chain. Why can this technique help increase the likelihood that w appears in the *j*th chain of $w_{j1}, \cdots, w_{jn_j}$?

1.10. Two readers of the first edition shared with us their experiences on distributing passwords:

- "I can recall a security incident where the user name and password were accidentally sent off the secure network to an unauthorized email address. While no further security incidents occurred, it was certainly possible for an attacker to recover the username and password and do serious damage to the network."
- "At work, we ONLY give passwords over the phone, and of course only when we know who we are speaking to. Of all the no-no's in network security, sending password via insecure emails has to be at the top of the list."

Describe your practice of distributing passwords and discuss their pros and cons.

1.11. "Early in my career as a Web developer," a reader of the first edition told us, "I created a Website for a friend. I created the FTP login name and password using the same first eight characters of the name of the site. In about 6 month time, somebody hacked into the site and put their own silly page in place of her content. Once I regained control, I created a high-strength password using a combination of uppercase and lowercase letters, numbers, and symbols, with a minimum of eight characters. I have since followed this practice for every Web login I create."

(a) Discuss what the Web developer did before being hacked was problematic.

(b) Do you think that the weak password the Web developer set up was the actual cause of his friend's computer being hacked? Justify your answer.

(c) Do you think that the Web developer's solution to the problem was effective? Justify your answer.

1.12. "Previously when I had DSL and an old router at home, the wireless encryption didn't work and I would occasionally find unauthorized users on my network," a reader told us. "I knew enough not to conduct any sensitive business using the wireless connection, but did once make an online shopping transaction using a credit card (I was being lazy). Within 2 days, there were fraudulent charges on my credit card." Make an educational guess what might happen and justify your answer.

1.13. "My account was compromised by a brute force attack a while back when I was playing an online game," said a reader of the first edition. "In response I purchased an RSA token and linked my account to it, so that even if my password was compromised again my account could never be fully accessed without the token code."

(a) Discuss why playing an online game might breach user accounts.

(b) Research the use of RSA tokens and explain whether using an RSA token would help secure user accounts for playing online games. Justify your answer.

1.14. A reader of the first edition reported the following social engineering attack happened to him: "Sometime ago I received a random phone call from someone (later identified as a fraudster) who wanted to speak to a senior person in my company.

Caller: Hello. Can I speak with the head of operations? (The fraudster did not mention a name, just a common job title, trying to sniff out a name and email address from me if I mistakenly mentioned the name of the person.)

Me: Can you please mention the name of the person you intend to reach, as we have many operation departments and heads around here (Baiting the fraudster)?

Caller: I have lost the business card he gave me and can't remember the details. Can you be kind enough to give me the name, email address, or direct number of one of the heads who might likely be in the same business meeting where I met the person I am trying to reach?

At this point the caller was suspicious enough that I transferred the call to my company's security investigative unit, which took it up from there."

(a) Describe whether you have a similar procedure at work and how you think the procedure could be improved.

(b) If you receive similar phone calls at home, what would you and should you do? Note that some crooks may call you that your tax returns contained errors and you must call a certain number to clear it up; otherwise you will be in trouble.

Others may change the content a little by, for example, telling you that your neighbors reported to the police department that you did something wrong. Anyway, all they try to get you to do is to call a certain number and then scare you to death so that you would provide them information or give them money.

1.15. Good baits are essential for a phishing attack to be successful. Baits are often presented in the form of email messages and Websites that appear to be authoritative. Links contained in phishing messages are traps, leading to Websites controlled by attackers. Discuss how to identify phishing messages and phishing sites.

1.16. The following phishing attacks were experienced by some of the readers. In each instance, describe what you would do if it happened to you.

(a) "A few years ago one of my network passwords on LinkedIn was compromised, possibly through phishing or pharming. As a result, spam messages spoofing my identity were sent to my connections. I discovered this when some of my connections notified me and said that they knew that I would not send such messages. I changed my passwords (and continue to do so periodically) and as a result the problem has not occurred since."

(b) "I received phishing emails 2 months ago (around November 2013), claiming to be from FedEX. There were several clues that they were bogus. For example, the content and the Subject Line did not look right, and nowhere did I see anything similar to `fedex.com`. The message was very generic about some complication in delivery, and it urged the recipient to open up a file attachment that looked very suspicious. Sometimes you can tell an email is a phishing attack because the link it gives you in the message does not look right."

(c) "I have been getting attacked very frequently through emails lately (in early 2014). One example is an email stating that I was offered a job, and asked me to fill out a form with all of my personal data. This is obviously an attempt to get my personal information because legitimate employers wouldn't offer me a job if they didn't know anything about me. My solution to the phishing attacks are never to login to anything through an email, and never giving out information to anyone I can't authenticate or trust. I think one of the main reasons that my phone number and email address were compromised is my resume being posted on sites like `monster.com`. As soon as I find a job I'm taking it down!"

(d) "I've received tons of phishing emails over the years. When I was a customer of a local bank, I encountered the best phishing email I have ever received. I received an email that looked like it was from the bank with a link to the Website. I clicked the link. When I was about to login, I noticed that the color of the site did not look right. I took a closer look at the URL, and realized that it was not the official Website of the bank. It almost tricked me. I blocked the sender and emailed the bank who then passed it along to the FBI."

(e) "I've encountered several cleverly disguised email invitations to provide account information. Thankfully, I've never entered personal information that was requested, but I know that many less security conscious people have. The best way to combat phishing is to ignore requests for personal information that emanate from the Web. When in doubt, call the institution directly, and not with the number on the email."

(f) “Just last week (i.e., in mid January 2014), I received a phishing email. It appeared to come from an organization I know, but the actual email address was obviously not, and contained (false) links to reset my password. I reported it to the IT Help Desk.”

1.17. Do you agree with the following rule of thumb when dealing with possible phishing emails: “If an email comes from a company or individual I don’t recognize, I delete it. If it’s really important, they will call me!” Justify your answer.

***1.18.** ARP maps an IP address to a MAC address of a computer. Thus, assigning a different MAC address to an IP address redirects message to a different computer. Conduct the following experiment. Let A, B, and C be three PCs connected to the same local area network (LAN) running Microsoft Windows (or Linux). Suppose that you have an user account on each of these computers and you have the same user name `fool` on computers B and C. Suppose that you can modify the ARP table on computer B (e.g., such as what a super user may do). On computer C, run `arp -a` to obtain its MAC address. Then on computer B, run `arp -s` to modify its ARP table to map B’s IP address to C’s MAC address. Wait for a while or reboot B to let B’s new ARP table take effect. Now, send an email message from your account on computer A to your account `fool` on computer B. This message will be redirected to your account `fool` on computer C. Verify this result in your experiment.

1.19. Use port scans to check your computer’s open ports.

(a) Use `ShieldsUP!` to scan your computer’s open ports for possible loopholes. Visit `www.grc.com` and click the `ShieldsUP!` link. Then move your mouse down to find the `ShieldsUP!` link. Click the link and follow the instructions to scan your computer’s open ports.

(b) `Nessus` has features similar to `ShieldsUP!`. It checks open ports and tries to determine what programs are running on them. Visit `www.nessus.org` and download `nessus`. Next, use `nessus` to scan your computer.

1.20. “Port scans are very frequent on our network by outside and inside attackers,” a reader told us. “We simply block repeat offenders.” Argue that this is a good solution. Can you think of a better approach to counter port scans? Justify your answers.

1.21. Web servers are easy targets of DoS attacks. For example, attackers may bombard a Web server with a large number of login attempts in a short period of time, forcing the Web server to use up its computing resources for checking passwords.

Web servers may use a picture verification service as follows: when receiving a login request, the Website opens a login page that will display, in addition to the usual windows for entering user name and password, a few characters in different colors or shapes, embedded in a small frame of colorful background and a window to enter these characters. To complete the login procedure, the user must also type in these characters. If these characters are not entered correctly, the Web server will not proceed to check the user name and password. This mechanism is typically used to prevent automation of services the Website provides and level the playing field (e.g., Ticketmaster uses this service to prevent scalpers from using a program to purchase tickets).

Explain how automation of services could be used to launch DoS attacks, and why the picture-verification mechanism may help stop DoS attacks.

1.22. A reader of the first edition shared this experience with us: "I sometimes saw employees bringing in a small personal switch and connecting it to the company LAN. Occasionally these switches would cause broadcast storms that resulted in denial of service on the LAN. It was easy to find these switches using tools such as wireshark and then remove them." These are rogue switches. Explain how to use wireshark to identify rogue switches.

1.23. "We had experienced repeated DoS attacks on our corporate Web servers," a reader said. "The attackers were flooding our servers with external communication requests, so much so that the servers could not respond to legitimate traffic. To counter these attacks, we moved to a SaaS solution for our online customer software from AWS (Amazon Web Services), and transitioned to a similar model for our corporate Web servers using a Rackspace provider, beefing up its security and redundancy during the transition."

(a) Conduct a research on AWS, SaaS, and Rackspace.

(b) On the basis of your research, argue that the solution the company took is a good one.

1.24. Sometimes, a legitimate application may affect the performance of your system. Googlebot, for example, is such an application. It is a highly debatable issue whether such applications are considered malware. Googlebot is a Web crawling tool developed by Google, which is also referred to as spider. It is used to crawl the Internet and discover new and updated pages for the Google index. Here is a story shared by a reader: "I worked with a customer who was facing extremely slow performance in their portal at the time of open enrollment for a new service. It was identified that it was Googlebot causing the problem, which was crawling the content on their external facing portal. They then worked with Google and the internal security team to filter the traffic to eliminate the additional crawling time."

Discuss this issue and justify your opinions.

1.25. Microsoft operating systems have become the household operating systems by people in all walks of life. Thus, computers that run Windows operating systems are hackers' major targets. Consequently, loopholes, flaws, and defects have been found one after another.

Use Microsoft Baseline Security Analyzer (MBSA) to analyze security settings of your Windows operating system and other Microsoft products. To do so, first download and install the newest version of MBSA from the following link:

`www.microsoft.com/technet/security/tools/mbsahome.mspx`

Then execute MBSA to scan your Windows system.

1.26. Server programs that run in the background of your computer are entry points to your computer from the network. Some of these programs are necessary, some are not, and some are malicious programs downloaded by careless users. Suppose that you are running Windows XP on your computer.

(a) Follow the following procedure to identify which server programs are running and which server programs have been closed: Select `Run` from the `Start` menu, then type in `msconfg`. Press the `OK` button to open the window of `System Configuration Utility`, and click `Services`. For example, is your `DHCP` client running or stopped?

(b) Follow the following steps to find out the usages of XP-supported services: Select `Run` from the `Start` menu, then type in `services.msc`. Press the `OK` bottom to open the window of `Services` and select `Services`. Select each service one at a time and read about its usage. For example, what is the usage of the `DHCP` client?

1.27. Back Orifice is a computer program designed for remote system administration to control a computer running the Windows operating system from a remote location. But it may also be use to log keystrokes easily. Other key-logging tools include hardware keylogger and invisible keylogger. Conduct a survey on keyloggers and write a paper reporting your findings.

1.28. Critical information may be stolen when you shop online. A reader shared with us the following story: "Just last year (i.e., in 2013) I had my credit card information stolen from what I believed to be a keystroke-logging attack. Since then I've beefed up my security and installed an anti-keylogger."

Identify and discuss security vulnerabilities you can think of associated with online shopping.

1.29. As we mentioned in the text, an apparently well-protected network could be brought down via an apparently minor trick. The following is a story shared by a reader of the first edition: "I am a system administrator for a large company with employees worldwide. My site produces sensitive hardware and software products. We have a very strong network security team keeping our network safe. However, about 2 years ago (i.e., in 2012), espionage hackers still managed to get into our network. As secure as our network was, the hackers used Outlook Web Access (OWA) to get into our network, retrieving a large volume of data in 2 days. The attack took the following steps:

1. They first collected information form media and by calling the company disguised as a sales person or government authority. They managed to retrieve email addresses from local users who were assigned to my site.
2. They then used a spoofing method to send emails to users from the known employees to other employees.
3. They would send emails with Trojans only during off hours, so that the email recipient would use OWA at home to access their email and bypass the firewalls and network security protocols at work.

The email spoofing was being done for about 2 weeks until a employee replied to the hacker, thinking it was an employee from a company laptop off hours. When the employee returned to the office the next day the hacker was able to bypass the firewall and get into the network. We had to make major changes to the network from top down including the following:

1. Removed all OWA installations.

2. Spent a large sum of money to purchase firewalls and network security devices and distributed them globally.
3. Hired ten additional network security professionals.
4. Removed all local administrative rights from domain computers.
5. Purchased USB token devices to key staff members with administrator rights on computers. The devise was a custom token that had both a certificate embedded in it associated with the employee and a password management code. For example, SafeNet. Inc sells such products (see `http://www.safenet-inc.com/data-protection/authentication/pki-authentication/`).
6. Required all employees to change passwords every 25 days for a year.

Also as a result of this attack, I had to travel for about a year to multiple locations two to three times a month to give network security training to users. We have not been hacked again so far and we continue to make improvements on our network. We send out intentional spoofing emails every now and then to test our employees and I have to give remote training to those who fail the tests."

(a) Discuss the attacking techniques the attacker was using in this attack.
(b) Discuss how to identify spoofing emails.

1.30. "Since the MafiaBoy attack in 2000, on a regular basis, our own servers have been hit by DDoS attacks on average once every 2 years," said a reader. Have you experienced any DDoS attack? If so, what measures did you take to counter DDoS attacks?

1.31. "I have discovered that DDoS effects can occur by accident on an alarming rate due to improperly configured application software. It is helpful if the network system is configured to shut down the troubled application. Otherwise, it can be difficult to use diagnostic tools to find it." Discuss how you may configure the system to detect misconfigured applications to address this reader's concern.

1.32. "Our servers were taken down with the Code Red and copycat worms in the early 2000s." The reader who shared this experienced also made the following comments: "Everybody I know has suffered from malicious software attacks at one time or another–no matter how careful you are. If you are not completely protected with updated anti-virus/malware software and more importantly, safe browsing habits, it can happen again to almost anyone."

Share your own experiences using one or more concrete examples of malicious software attacks you encountered.

1.33. "Several lab computers I administered were infected with viruses that hijacked the system," a reader told us. "The infected system displayed a message supposedly from the FBI saying that the system was in violation of copyright laws and for a small fee could be cleared up (using a credit card of course). It frankly was too much work to clean it up so we instead just reinstalled the system."

Suggest a way to cleanup such viruses without reinstalling the system. Justify your answer.

1.34. "Back at the dawn of time when I was an undergrad," said a reader, "my university's computers were riddled with viruses. One that I remember in particular was the Stoned virus. It would attack the file allocation table in the DOS operating system, making the computer unable to find any file. Antivirus tools were not readily available then, so I kept a floppy disk that was just for the university computers. Once I used a disk at school, I marked it and never used it anywhere else. I'm sure that helped spread the virus on the university computers but it kept the viruses off my own PC."

While floppy disks are no longer in use, USB sticks are still widely used today. How do you like the reader's approach to viruses and justify your answer.

1.35. "In early 2013 I built a Website for a local restaurant using Drupal. It was a relatively straightforward site, with no actual commerce function. It didn't have any personal information on it, or in the MySQL database back end. I hid the administrative login for Drupal, but not very well. I just put it somewhere where a site user couldn't navigate to. However, Drupal is set up in such a way that site structures can be guessed by hackers, or perhaps mine was just crawled somehow by a program specializing in this sort of thing. Almost every day I received requests to add users to the site. The restaurant went out of business last week, so I took the site down, which stopped the requests right away."

Can you suggest what happened to the Website and offer a fix if the site were to be run?

1.36. "This past year (e.g., in 2013), I developed a quick and easy site for one of our meetings on a subdomain especially for it outside of our usual security model. One morning, my inbox was flooded with hundreds of error messages (i.e., error messages sent from sites to developers with all the parameters of the requests), all with SQL statements embedded in an open text field's input string. Fortunately, none of the attempts to access the database was successful and that day we came up with a procedure to prevent it from happening in the future by (1) validating all input before it is submitted and (2) blocking any suspicious statements before they get submitted to the database."

Describe how to identify suspicious SQL statements.

1.37. "A few months ago in 2013, a coworker of mine turned on an old PC hooked up to our work network and did not tell anyone. That old PC had been off line for a couple of years. Within a day or two we were having all kinds of network problems, from performance slowdown to other weird issues. Because this PC was behind our firewall it was not picked up right away. It turned out that all these problems were caused simply by an old worm in that old PC. To remedy the situation we first removed that old PC. We then manually scanned all our PCs and servers with multiple antivirus and malware tools, for the worm had also compromised the antivirus software installed on the PCs. We shut down the ports and services the worm was spreading through until we were sure that the network was clean. Once clean we were able to reconnect everything and went back to business as usual. This took about 72 hours to remedy. This incident made us revise our security policies and procedures to prevent things like this from happening again."

What do you think the new security policy should be for this reader's company to avoid similar incidents mentioned in the message from happening again?

1.38. Junk email filters are software tools used to prevent junk email messages from entering your mailbox. Microsoft Office Outlook has this feature. To set it up, open Office Outlook and click `Actions`. Point the mouse to `Junk E-Mail`, click `Junk E-mail options`, `Safe Lists Only`, `Safe Senders`, and `Add`. Type in here the email addresses you wish to receive email messages from, then click `OK`. Likewise, you may also specify the email addresses that you do not want to receive messages from. Describe how this can be done.

1.39. "A server I managed was once compromised by an attacker. The attacker gained root access using buffer overflow and installed a Trojan that replaced standard Linux commands with infected ones, opening up ports for the attacker to attack other locations. We fixed the problem by a complete system reinstall from original media and applied proper security patches."

Describe what each of these two remedies do.

1.40. **Canary Values.** The GNU Compiler Collection (GCC) supports buffer overflow protection using random canary values.

(a) Determine what the `-fstack-protector` and `-fstack-protector-all` flags are used for when compiling code using the GNU C (`gcc`) and C++ (`g++`) compilers.

(b) Compile C code with and without the `-fstack-protector-all` flag and disassemble the executables using the Linux tool `objdump`, with the `-d` option, compare the output and determine what code is responsible for inserting the canary value in the prologue and what code is responsible for checking the canary value in the epilogue.

1.41. "When I was a kid I had problems with adware and Trojans on my Windows PC. Since then I always make sure that my machines have security software installed. Now I am using Norton Internet Security and it seems to get the job done. We also have Norton endpoint security installed on the development VMs at work." Have you experienced any malicious software attack that even the Norton security tools did not help remove them?

1.42. "I had an infection with spyware on my home computer. It popped up a window with an instruction to download Windows antivirus software. It would popup and keep popping up until my computer would freeze because all the opened windows had used up all of the memory. I looked up how to fix the problem but it seemed so involved I finally just took the easy route: I wiped my hard drive, reinstalled the system from scratch, and downloaded antivirus and spyware tools. On a separate note, in my work we use common access cards to log in to computers and we can't even plug in a USB for fear that there might be malware on it."

(a) What do you think happened to this person's computer?

(b) Is the USB policy mentioned a good policy? Justify your answer.

1.43. "I have had several instances where my wife's computer became infected with some form of malware or another. She visited several questionable sites that I cautioned her against, but the joy of those sites outweighed my warnings. Of course each time her computer was infected I would have to fix it and hear about why can't I stop her machine from being infected. To help me avoid this, I run Linux at home which I have found to be much more secure, and less susceptible to viruses."

Do you agree with this reader's last comment about Linux being much more secure and less susceptible to viruses? Justify your answers.

1.44. "I once worked for a guy as a consultant," a reader told us. "The guy started bragging about the logic bomb he created. He set things up so that 3 months after he left the company (due to downsizing), the company screens would be taken over by a faked video of a senior member of the management team having inappropriate relationships with a donkey. The company then called him back in as a consultant (since he knew the system so well) to help find the cause of the problem. He worked at a very high rate of pay for 4 months pretending to solve the problem created by him. I stopped working for him the next week."

Are you aware of any person who planned logic bombs in the software they wrote? If so, please describe it. If not, imagine and describe a situation in which a logic bomb may be planned.

1.45. "In 2012, some syndicates were able to hack into our credit-card payment systems in North America, causing us financial loss of up to $2.7 million dollars. They did this through a combination of password theft, cryptanalysis, and phishing emails. Like the textbook says: The battle against network attacks is a perpetual one as the various attackers constantly device new means to breach our network securities." Can you make an educational guess when the attack this reader mentioned might take place? Justify your answers.

****1.46.** When the TCP/IP protocols and the OSI seven-layer model were devised, their designers were only concerned about how to efficiently and reliably transmit data from the source computer to the destination computer. Data security was not a concern at that time. Consequently, the TCP/IP protocols and the OSI model do not contain any built-in security mechanism. When they later realized this security weakness, protocol designers started to add all kinds of security mechanisms into communication protocols. But these early protocols were not designed for data security, and so they may not have the right framework for adding security features. Adding a security feature to a protocol not built for it is like taking out materials from a wall to mend a fence. Thus, network designers have started to investigate the following issue: if one designs a communication protocol all over again, what would be the best native architecture for including the current security mechanisms as well as for adding future security features. Think about this issue when you read the rest of the book, and try to develop a design of your own. This exercise is to be handed in at the end of the course.