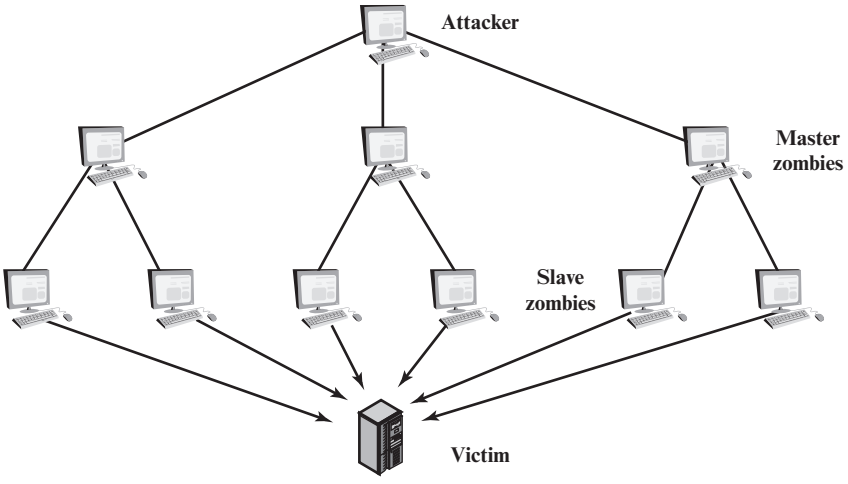
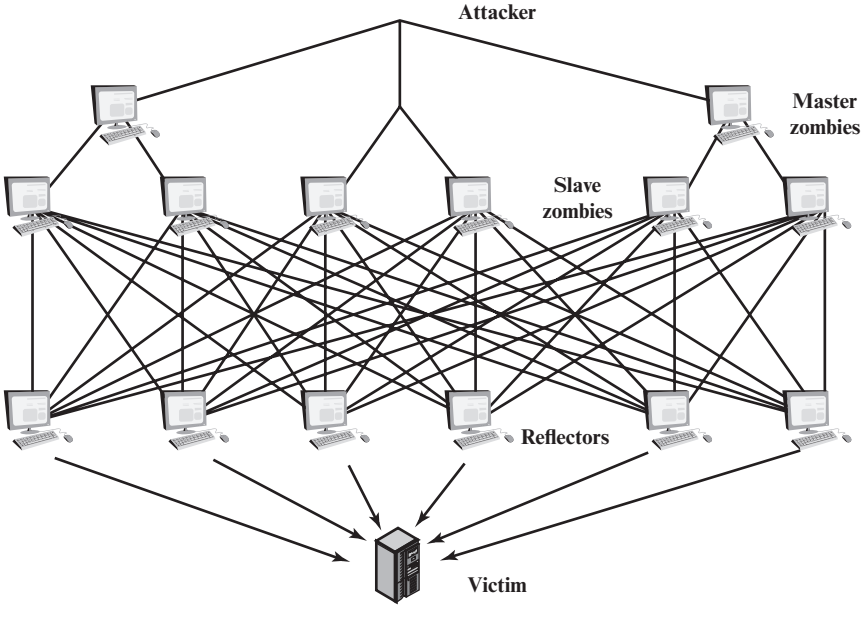


(a) Direct DDoS Attack



(b) Reflector DDoS Attack

Figure 10.6 Types of Flooding-Based DDoS Attacks

Constructing the Attack Network

The first step in a DDoS attack is for the attacker to infect a number of machines with zombie software that will ultimately be used to carry out the attack. The essential ingredients in this phase of the attack are the following:

1. Software that can carry out the DDoS attack. The software must be able to run on a large number of machines, must be able to conceal its existence, must

be able to communicate with the attacker or have some sort of time-triggered mechanism, and must be able to launch the intended attack toward the target.

2. A vulnerability in a large number of systems. The attacker must become aware of a vulnerability that many system administrators and individual users have failed to patch and that enables the attacker to install the zombie software.
3. A strategy for locating vulnerable machines, a process known as scanning.

In the scanning process, the attacker first seeks out a number of vulnerable machines and infects them. Then, typically, the zombie software that is installed in the infected machines repeats the same scanning process, until a large distributed network of infected machines is created. [MIRK04] lists the following types of scanning strategies:

- **Random:** Each compromised host probes random addresses in the IP address space, using a different seed. This technique produces a high volume of Internet traffic, which may cause generalized disruption even before the actual attack is launched.
- **Hit list:** The attacker first compiles a long list of potential vulnerable machines. This can be a slow process done over a long period to avoid detection that an attack is underway. Once the list is compiled, the attacker begins infecting machines on the list. Each infected machine is provided with a portion of the list to scan. This strategy results in a very short scanning period, which may make it difficult to detect that infection is taking place.
- **Topological:** This method uses information contained on an infected victim machine to find more hosts to scan.
- **Local subnet:** If a host is infected behind a firewall, that host then looks for targets in its own local network. The host uses the subnet address structure to find other hosts that would otherwise be protected by the firewall.

DDoS Countermeasures

In general, there are three lines of defense against DDoS attacks [CHAN02]:

- **Attack prevention and preemption (before the attack):** These mechanisms enable the victim to endure attack attempts without denying service to legitimate clients. Techniques include enforcing policies for resource consumption and providing backup resources available on demand. In addition, prevention mechanisms modify systems and protocols on the Internet to reduce the possibility of DDoS attacks.
- **Attack detection and filtering (during the attack):** These mechanisms attempt to detect the attack as it begins and respond immediately. This minimizes the impact of the attack on the target. Detection involves looking for suspicious patterns of behavior. Response involves filtering out packets likely to be part of the attack.
- **Attack source traceback and identification (during and after the attack):** This is an attempt to identify the source of the attack as a first step in preventing future attacks. However, this method typically does not yield results fast enough, if at all, to mitigate an ongoing attack.

The challenge in coping with DDoS attacks is the sheer number of ways in which they can operate. Thus, DDoS countermeasures must evolve with the threat.

10.12 KEY TERMS, REVIEW QUESTIONS, AND PROBLEMS

Key Terms

adware	e-mail virus	reflector DDoS attack
attack kit	flooders	rootkit
backdoor	keyloggers	scanning
behavior-blocking software	logic bomb	spear-phishing
blended attack	macro virus	spyware
boot sector infector	malicious software	stealth virus
bot	malware	trapdoor
botnet	metamorphic virus	Trojan horse
crimeware	mobile code	virus
direct DDoS attack	parasitic virus	worm
distributed denial of service (DDoS)	phishing	zombie
	polymorphic virus	zero-day exploit
downloader	ransomware	
drive-by-download		

Review Questions

10.1 What are three broad mechanisms that malware can use to propagate?
10.2 What is a blended attack?
10.3 What are typical phases of operation of a virus or worm?
10.4 Classify viruses based on the targets they try to infect.
10.5 List the features of macro viruses that enable them to infect scripting codes.
10.6 What functions does a worm perform during the propagation phase?
10.7 Give some examples of client side vulnerabilities that can be exploited by malware?
10.8 What is an "infection vector"?
10.9 Explain the difference between a keylogger and spyware with an example.
10.10 What kind of activities can be performed by an attacker using a rootkit? What makes it difficult to detect a rootkit?
10.11 Describe some malware countermeasure elements.
10.12 List three places malware mitigation mechanisms may be located.
10.13 Briefly describe the four generations of antivirus software.
10.14 List the activities that can be monitored by "behavior-blocking software".
10.15 What is the difference between a reflector DDoS attack and a direct DDoS attack?

Problems

10.1 There is a flaw in the virus program of Figure 10.1a. What is it?
10.2 The question arises as to whether it is possible to develop a program that can analyze a piece of software to determine if it is a virus. Consider that we have a program D

that is supposed to be able to do that. That is, for any program P, if we run D(P), the result returned is TRUE (P is a virus) or FALSE (P is not a virus). Now consider the following program:

```
Program CV :=
 { . . .
 main-program :=
      {if D(CV) then goto next:
            else infect-executable;
      }
 next:
 }
```

In the preceding program, infect-executable is a module that scans memory for executable programs and replicates itself in those programs. Determine if D can correctly decide whether CV is a virus.

10.3 The following code fragments show a sequence of virus instructions and a metamorphic version of the virus. Describe the effect produced by the metamorphic code.

Original Code	Metamorphic Code
`mov eax, 5`	`mov eax, 5`
`add eax, ebx`	`push edx`
`call [ebx]`	`jmp 0x89AB`
	`swap eax, ebx`
	`call [ebx]`
	`nop`

10.4 The list of passwords used by the Morris worm is provided at this book's Premium Content Web site.

a. The assumption has been expressed by many people that this list represents words commonly used as passwords. Does this seem likely? Justify your answer.

b. If the list does not reflect commonly used passwords, suggest some approaches that Morris may have used to construct the list.

10.5 What type of malware is the following code fragment?

```
legitimate code
if data is Friday the 13th;
    crash_computer();
legitimate code
```

10.6 Consider the following situation and identify the type of software attack, if any:

> You are the owner of a small business. After you login to your client server application with your credentials, you find that the data is displayed in the form of a jumbled collection of alphabets, numbers, special characters, and symbols. You are unpleasantly surprised and wonder what happened. You get a call after some time, and the person at the other end tells you that your system is hacked, and you can recover the data once you pay him a certain amount of money.

10.7 Assume that you have received an e-mail with an attachment from your friend's e-mail id. You access the e-mail using your work computer, and click on the

attachment without screening it for malware. What threats might this pose to your work computer?

10.8 Suppose you observe that your home PC is responding very slowly to information requests from the net. And then you further observe that your network gateway shows high levels of network activity, even though you have closed your e-mail client, Web browser, and other programs that access the net. What types of malware could cause these symptoms? Discuss how the malware might have gained access to your system. What steps can you take to check whether this has occurred? If you do identify malware on your PC, how can you restore it to safe operation?

10.9 Suppose while browsing the Internet, you get a popup window stating that you need to install this software in order to clean your system as it is running low on resources. Since the message seems to be from a genuine OS vendor like Microsoft Windows or Mac iOS, you click the 'OK' button. How could your action harm your system? How can you fix the issue?

10.10 Suppose you have a new smartphone and are excited about the range of apps available for it. You read about a really interesting new game that is available for your phone. You do a quick Web search for it and see that a version is available from one of the free marketplaces. When you download and start to install this app, you are asked to approve the access permissions granted to it. You see that it wants permission to "Send SMS messages" and to "Access your address-book." Should you be suspicious that a game wants these types of permissions? What threat might the app pose to your smartphone? Should you grant these permissions and proceed to install it? What types of malware might it be?

10.11 Assume you receive an e-mail that appears to come from a senior manager of your company, with a subject indicating that it concerns a project that you are currently working on. When you view the e-mail, you see that it asks you to review the attached revised press release, supplied as a PDF document, to check that all details are correct before management releases it. When you attempt to open the PDF, the viewer pops up a dialog labeled "Launch File," indicating that "the file and its viewer application are set to be launched by this PDF file." In the section of this dialog labeled "File" there are a number of blank lines and finally the text "Click the 'Open' button to view this document." You also note that there is a vertical scroll-bar visible for this region. What type of threat might this pose to your computer system should you indeed select the "Open" button? How could you check your suspicions without threatening your system? What type of attack is this type of message associated with? How many people are likely to have received this particular e-mail?

10.12 Assume you work in a financial auditing company. An e-mail arrives in your inbox that appears to be from your chief auditor with the following content:

> "We have identified a few threats which pose potential danger to our information systems. In order to address this, our information security team has decided to ensure proper credentials of all the employees. Please cooperate and complete this process immediately by clicking the given link."

What kind of an attack is this e-mail attempting? How should you respond to such e-mails?

10.13 There are hundreds of unsolicited e-mails in your inbox. What kind of attack is this? Analyze related issues.

10.14 Suggest some methods of attacking the worm countermeasure architecture, discussed in Section 10.9, that could be used by worm creators. Suggest some possible countermeasures to these methods.

CHAPTER 11

INTRUDERS

LEARNING OBJECTIVES

After studying this chapter, you should be able to:

- Distinguish among various types of intruder behavior patterns.
- Understand the basic principles of and requirements for intrusion detection.
- Discuss the key features of intrusion detection systems.
- Define the intrusion detection exchange format.
- Explain the purpose of honeypots.
- Explain the mechanism by which hashed passwords are used for user authentication.
- Understand the use of the Bloom filter in password management.

A significant security problem for networked systems is hostile, or at least unwanted, trespass by users or software. User trespass can take the form of unauthorized logon to a machine or, in the case of an authorized user, acquisition of privileges or performance of actions beyond those that have been authorized. Software trespass can take the form of a virus, worm, or Trojan horse.

All these attacks relate to network security because system entry can be achieved by means of a network. However, these attacks are not confined to network-based attacks. A user with access to a local terminal may attempt trespass without using an intermediate network. A virus or Trojan horse may be introduced into a system by means of an optical disc. Only the worm is a uniquely network phenomenon. Thus, system trespass is an area in which the concerns of network security and computer security overlap.

Because the focus of this book is network security, we do not attempt a comprehensive analysis of either the attacks or the security countermeasures related to system trespass. Instead, in this Part we present a broad overview of these concerns.

This chapter covers the subject of intruders. First, we examine the nature of the attack and then look at strategies intended for prevention and, failing that, detection. Next we examine the related topic of password management.

11.1 INTRUDERS

One of the two most publicized threats to security is the intruder (the other is viruses), often referred to as a hacker or cracker. In an important early study of intrusion, Anderson [ANDE80] identified three classes of intruders:

- **Masquerader:** An individual who is not authorized to use the computer and who penetrates a system's access controls to exploit a legitimate user's account

- **Misfeasor:** A legitimate user who accesses data, programs, or resources for which such access is not authorized, or who is authorized for such access but misuses his or her privileges
- **Clandestine user:** An individual who seizes supervisory control of the system and uses this control to evade auditing and access controls or to suppress audit collection

The masquerader is likely to be an outsider, the misfeasor generally is an insider, and the clandestine user can be either an outsider or an insider.

Intruder attacks range from the benign to the serious. At the benign end of the scale, there are many people who simply wish to explore internets and see what is out there. At the serious end are individuals who are attempting to read privileged data, perform unauthorized modifications to data, or disrupt the system.

[GRAN04] lists the following examples of intrusion:

- Performing a remote root compromise of an e-mail server
- Defacing a Web server
- Guessing and cracking passwords
- Copying a database containing credit card numbers
- Viewing sensitive data, including payroll records and medical information, without authorization
- Running a packet sniffer on a workstation to capture usernames and passwords
- Using a permission error on an anonymous FTP server to distribute pirated software and music files
- Dialing into an unsecured modem and gaining internal network access
- Posing as an executive, calling the help desk, resetting the executive's e-mail password, and learning the new password
- Using an unattended, logged-in workstation without permission

Intruder Behavior Patterns

The techniques and behavior patterns of intruders are constantly shifting, to exploit newly discovered weaknesses and to evade detection and countermeasures. Even so, intruders typically follow one of a number of recognizable behavior patterns, and these patterns typically differ from those of ordinary users. In the following, we look at three broad examples of intruder behavior patterns, to give the reader some feel for the challenge facing the security administrator.

Hackers Traditionally, those who hack into computers do so for the thrill of it or for status. The hacking community is a strong meritocracy in which status is determined by level of competence. Thus, attackers often look for targets of opportunity and then share the information with others. A typical example is a break-in at a large financial institution reported in [RADC04]. The intruder took advantage of the fact that the corporate network was running unprotected services, some of which were not even needed. In this case, the key to the break-in was the pcAnywhere application. The manufacturer, Symantec, advertises this program as a remote control

solution that enables secure connection to remote devices. But the attacker had an easy time gaining access to pcAnywhere; the administrator used the same three-letter username and password for the program. In this case, there was no intrusion detection system on the 700-node corporate network. The intruder was only discovered when a vice-president walked into her office and saw the cursor moving files around on her Windows workstation.

Benign intruders might be tolerable, although they do consume resources and may slow performance for legitimate users. However, there is no way in advance to know whether an intruder will be benign or malign. Consequently, even for systems with no particularly sensitive resources, there is a motivation to control this problem.

Intrusion detection systems (IDSs) and intrusion prevention systems (IPSs) are designed to counter this type of hacker threat. In addition to using such systems, organizations can consider restricting remote logons to specific IP addresses and/or use virtual private network technology.

One of the results of the growing awareness of the intruder problem has been the establishment of a number of computer emergency response teams (CERTs). These cooperative ventures collect information about system vulnerabilities and disseminate it to systems managers. Hackers also routinely read CERT reports. Thus, it is important for system administrators to quickly insert all software patches to discovered vulnerabilities. Unfortunately, given the complexity of many IT systems, and the rate at which patches are released, this is increasingly difficult to achieve without automated updating. Even then, there are problems caused by incompatibilities resulting from the updated software. Hence the need for multiple layers of defense in managing security threats to IT systems.

Criminals Organized groups of hackers have become a widespread and common threat to Internet-based systems. These groups can be in the employ of a corporation or government but often are loosely affiliated gangs of hackers. Typically, these gangs are young, often Eastern European, Russian, or southeast Asian hackers who do business on the Web [ANTE06]. They meet in underground forums with names like DarkMarket.org and theftservices.com to trade tips and data and coordinate attacks. A common target is a credit card file at an e-commerce server. Attackers attempt to gain root access. The card numbers are used by organized crime gangs to purchase expensive items and are then posted to carder sites, where others can access and use the account numbers; this obscures usage patterns and complicates investigation.

Whereas traditional hackers look for targets of opportunity, criminal hackers usually have specific targets, or at least classes of targets in mind. Once a site is penetrated, the attacker acts quickly, scooping up as much valuable information as possible and exiting.

IDSs and IPSs can also be used for these types of attackers, but may be less effective because of the quick in-and-out nature of the attack. For e-commerce sites, database encryption should be used for sensitive customer information, especially credit cards. For hosted e-commerce sites (provided by an outsider service), the e-commerce organization should make use of a dedicated server (not used to support multiple customers) and closely monitor the provider's security services.

Insider Attacks Insider attacks are among the most difficult to detect and prevent. Employees already have access and knowledge about the structure and content of corporate databases. Insider attacks can be motivated by revenge or simply a feeling of entitlement. An example of the former is the case of Kenneth Patterson, fired from his position as data communications manager for American Eagle Outfitters. Patterson disabled the company's ability to process credit card purchases during five days of the holiday season of 2002. As for a sense of entitlement, there have always been many employees who felt entitled to take extra office supplies for home use, but this now extends to corporate data. An example is that of a vice-president of sales for a stock analysis firm who quit to go to a competitor. Before she left, she copied the customer database to take with her. The offender reported feeling no animus toward her former employee; she simply wanted the data because it would be useful to her.

Although IDS and IPS facilities can be useful in countering insider attacks, other more direct approaches are of higher priority. Examples include the following:

- Enforce least privilege, only allowing access to the resources employees need to do their job.
- Set logs to see what users access and what commands they are entering.
- Protect sensitive resources with strong authentication.
- Upon termination, delete employee's computer and network access.
- Upon termination, make a mirror image of employee's hard drive before reissuing it. That evidence might be needed if your company information turns up at a competitor.

In this section, we look at the techniques used for intrusion. Then we examine ways to detect intrusion.

Intrusion Techniques

The objective of the intruder is to gain access to a system or to increase the range of privileges accessible on a system. Most initial attacks use system or software vulnerabilities that allow a user to execute code that opens a backdoor into the system. Alternatively, the intruder attempts to acquire information that should have been protected. In some cases, this information is in the form of a user password. With knowledge of some other user's password, an intruder can log in to a system and exercise all the privileges accorded to the legitimate user.

Typically, a system must maintain a file that associates a password with each authorized user. If such a file is stored with no protection, then it is an easy matter to gain access to it and learn passwords. The password file can be protected in one of two ways:

- **One-way function:** The system stores only the value of a function based on the user's password. When the user presents a password, the system transforms that password and compares it with the stored value. In practice, the system usually performs a one-way transformation (not reversible), in which the password is used to generate a key for the one-way function and in which a fixed-length output is produced.
- **Access control:** Access to the password file is limited to one or a very few accounts.