CompTIA SY0-201
Security+

Exam Guide

Version 2.0

Leading The Way
in IT Testing And Certification Tools

www.testking.com

Table of Contents

xiii. IDEA
xiv. Symmetric Encryption Summary
xv. Asymmetric Encryption
xvi. RSA
xvii. Diffie-Hellman
xviii. ElGamal
xix. ECC
xx. Asymmetric Encryption Summary
xxi. Steganography
xxii. Cryptography Algorithm Use
xxiii. Confidentiality
xxiv. Integrity
xxv. Nonrepudiation
xxvi. Authentication
xxvii. Digital Signatures
xxviii. Key Escrow
xxix. Cryptographic Applications

- **Public Key Infrastructure**
  i. The Basics of Public Key Infrastructures
  ii. Certificate Authorities
  iii. Registration Authorities
  iv. Local Registration Authorities
  v. Certificate Repositories
  vi. Trust and Certificate Verification
  vii. Digital Certificates
  viii. Certificate Attributes
  ix. Certificate Extensions
  x. Certificate Lifecycles
  xi. Centralized or Decentralized Infrastructures
  xii. Hardware Storage Devices
  xiii. Private Key Protection
  xiv. Key Recovery
  xv. Key Escrow
  xvi. Public Certificate Authorities
  xvii. In-house Certificate Authorities
  xviii. Outsourced Certificate Authorities
- **Security in Infrastructure**
  i. Physical Security
  ii. The Security Problem
  iii. Physical Security Safeguards
    1. Walls and Guards
    2. Policies and Procedures
    3. Access Controls and Monitoring
    4. Environmental Controls
    5. Authentication

iv. Infrastructure Security
    1. Devices
    2. Workstations
    3. Servers
    4. Network Interface Cards
    5. Hubs
    6. Bridges
    7. Switches
    8. Routers
    9. Firewalls
    10. Wireless
    11. Modems
    12. Telecom/PBX
    13. RAS
    14. VPN
    15. Intrusion Detection Systems
    16. Network Access Control
    17. Network Monitoring/Diagnostic
    18. Mobile Devices
v. Media
    1. Coaxial Cable
    2. UTP/STP
    3. Fiber
    4. Unguided Media
vi. Security Concerns for Transmission Media
    1. Physical Security
vii. Removable Media
    1. Magnetic Media
    2. Optical Media
    3. Electronic Media
viii. Security Topologies
    1. Security Zones
    2. Telephony
    3. VLANs
    4. NAT
ix. Tunneling

- Security in Transmissions
    i. Intrusion Detection Systems
    ii. History of Intrusion Detection Systems
    iii. IDS Overview
    iv. Host-based IDSs
        1. Advantages of HIDSs
        2. Disadvantages of HIDSs
        3. Active vs. Passive HIDSs
        4. Resurgence and Advancement of HIDSs

v. PC-based Malware Protection
    1. Antivirus Products
    2. Personal Software Firewalls
    3. Pop-up Blocker
    4. Windows Defender
vi. Network-based IDSs
    1. Advantages of a NIDS
    2. Disadvantages of a NIDS
    3. Active vs. Passive NIDSs
vii. Signatures
viii. False Positives and Negatives
ix. IDS Models
x. Intrusion Prevention Systems
xi. Honeypots and Honeynets
xii. Firewalls
xiii. Proxy Servers
xiv. Internet Content Filters
xv. Protocol Analyzers
xvi. Network Mappers
xvii. Anti-spam

- Types of Attacks and Malicious Software
    i. Avenues of Attack.
        1. The Steps in an Attack
        2. Minimizing Possible Avenues of Attack
    ii. Attacking Computer Systems and Networks
        1. Denial-of-Service Attacks
        2. Backdoors and Trapdoors
        3. Null Sessions
        4. Sniffing
        5. Spoofing
        6. Man-in-the-Middle Attacks
        7. Replay Attacks
        8. TCP/IP Hijacking
        9. Attacks on Encryption
        10. Address System Attacks
        11. Password Guessing
        12. Software Exploitation
        13. Malicious Code
        14. War-Dialing and War-Driving
        15. Social Engineering
    iii. Auditing
- Web Components
- Current Web Components and Concerns
- Protocols
    i. Encryption (SSL and TLS)

ii. The Web (HTTP and HTTPS)
iii. Directory Services (DAP and LDAP)
iv. File Transfer (FTP and SFTP)
v. Vulnerabilities

- Code-Based Vulnerabilities
  i. Buffer Overflows
  ii. Java and JavaScript
  iii. ActiveX
  iv. Securing the Browser
  v. CGI
  vi. Server-Side Scripts
  vii. Cookies
  viii. Signed Applets
  ix. Browser Plug-ins
- Application-Based Weaknesses
  i. Open Vulnerability and Assessment Language (OVAL)

Security Concepts

General Security Concepts

Basic Security Terminology

The term hacking is used frequently in the media. A hacker was once considered an individual who understood the technical aspects of computer operating systems and networks. Hackers were individuals you turned to when you had a problem and needed extreme technical expertise. Today, as a result of the media use, the term is used more often to refer to individuals who attempt to gain unauthorized access to computer systems or networks. While some would prefer to use the terms cracker and cracking when referring to this nefarious type of activity, the terminology generally accepted by the public is that of hacker and hacking. A related term that is sometimes used is phreaking, which refers to the "hacking" of computers and systems used by the telephone company.

Security Basics

Computer security is a term that has many meanings and related terms. Computer security entails the methods used to ensure that a system is secure. The ability to control who has access to a computer system and data and what they can do with those resources must be addressed in broad terms of computer security.

Seldom in today's world are computers not connected to other computers in networks. This then introduces the term network security to refer to the protection of the multiple computers and other devices that are connected together in a network. Related to these two terms are two others, information security and information assurance, which place the focus of the security process not on the hardware and software being used but on the data that is processed by them. Assurance also introduces another concept, that of the availability of the systems and information when users want them.

Since the late 1990s, much has been published about specific lapses in security that has resulted in the penetration of a computer network or in denying access to or the use of the network. Over the last few years, the general public has become increasingly aware of its

dependence on computers and networks and consequently has also become interested in their security.

As a result of this increased attention by the public, several new terms have become commonplace in conversations and print. Terms such as hacking, virus, TCP/IP, encryption, and firewalls now frequently appear in mainstream news publications and have found their way into casual conversations. What was once the purview of scientists and engineers is now part of our everyday life.

With our increased daily dependence on computers and networks to conduct everything from making purchases at our local grocery store to driving our children to school (any new car these days probably uses a small computer to obtain peak engine performance), ensuring that computers and networks are secure has become of paramount importance. Medical information about each of us is probably stored in a computer somewhere. So is financial information and data relating to the types of purchases we make and store preferences (assuming we have and use a credit card to make purchases).

Making sure that this information remains private is a growing concern to the general public, and it is one of the jobs of security to help with the protection of our privacy. Simply stated, computer and network security is essential for us to function effectively and safely in today's highly automated environment.

The "CIA" of Security

Almost from its inception, the goals of computer security have been threefold: confidentiality, integrity, and availability—the "CIA" of security. Confidentiality ensures that only those individuals who have the authority to view a piece of information may do so. No unauthorized individual should ever be able to view data to which they are not entitled. Integrity is a related concept but deals with the modification of data. Only authorized individuals should be able to change or delete information. The goal of availability is to ensure that the data, or the system itself, is available for use when the authorized user wants it.

As a result of the increased use of networks for commerce, two additional security goals have been added to the original three in the CIA of security. Authentication deals with ensuring that an individual is who he claims to be. The need for authentication in an online banking transaction, for example, is obvious. Related to this is nonrepudiation, which deals with the ability to verify that a message has been sent and received so that the sender (or receiver) cannot refute sending (or receiving) the information.

The Operational Model of Security

For many years, the focus of security was on prevention. If you could prevent somebody from gaining access to your computer systems and networks, you assumed that they were secure. Protection was thus equated with prevention. While this basic premise was true, it failed to

acknowledge the realities of the networked environment of which our systems are a part. No matter how well you think you can provide prevention, somebody always seems to find a way around the safeguards. When this happens, the system is left unprotected. What is needed is multiple prevention techniques and also technology to alert you when prevention has failed and to provide ways to address the problem.

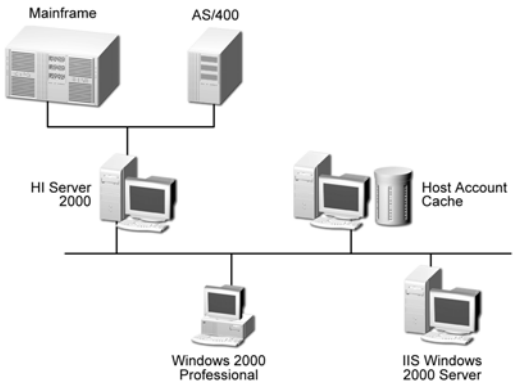
This results in a modification to the original security equation with the addition of two new elements - detection and response. The security equation thus becomes Protection = Prevention + (Detection + Response). This is known as the operational model of computer security. Every security technique and technology falls into at least one of the three elements of the equation.

Security Basics

An organization can choose to address the protection of its networks in three ways: ignore security issues, provide host security, and approach security at a network level. The last two, host and network security, have prevention as well as detection and response components.

If an organization decides to ignore security, it has chosen to utilize the minimal amount of security that is provided with its workstations, servers, and devices. No additional security measures will be implemented. Each "out-of-the-box" system has certain security settings that can be configured, and they should be. To protect an entire network, however, requires work in addition to the few protection mechanisms that come with systems by default.

Host Security



Host security takes a granular view of security by focusing on protecting each computer and device individually instead of addressing protection of the network as a whole. When host security is implemented, each computer is expected to protect itself. If an organization decides to implement only host security and does not include network security, it will likely introduce or overlook vulnerabilities. Many environments involve different operating systems (Windows, UNIX, Linux, and Macintosh), different versions of those operating systems, and different types of installed applications. Each operating system has security configurations that differ from other systems, and different versions of the same operating system can in fact have variations among them. Trying to ensure that every computer is "locked down" to the same degree as every other system in the environment can be overwhelming and often results in an unsuccessful and frustrating effort.

Network Security
In some smaller environments, host security alone might be a viable option, but as systems become connected into networks, security should include the actual network itself. In network security, an emphasis is placed on controlling access to internal computers from external entities. This control can be through devices such as routers, firewalls, authentication hardware and software, encryption, and intrusion detection systems (IDSs).

Least Privilege
One of the most fundamental approaches to security is least privilege. This concept is applicable to many physical environments as well as network and host security. Least privilege means that an object (such as a user, application, or process) should have only the rights and privileges necessary to perform its task, with no additional permissions. Limiting an object's privileges limits the amount of harm that can be caused, thus limiting an organization's exposure to damage. Users may have access to the files on their workstations and a select set of files on a file server, but they have no access to critical data that is held within the database. This rule helps an organization protect its most sensitive resources and helps ensure that whoever is interacting with these resources has a valid reason to do so.

The concept of least privilege applies to more network security issues than just providing users with specific rights and permissions. When trust relationships are created, they should not be implemented in such a way that everyone trusts each other simply because it is easier to set it up that way. One domain should trust another for very specific reasons, and the implementers should have a full understanding of what the trust relationship allows between two domains. If one domain trusts another, do all of the users automatically become trusted, and can they thus easily access any and all resources on the other domain? Is this a good idea? Can a more secure method provide the same functionality? If a trusted relationship is implemented such that users in one group can access a plotter or printer that is available on only one domain, for example, it might make sense to purchase another plotter so that other more valuable or sensitive resources are not accessible by the entire group.

Separation of Duties
Another fundamental approach to security is separation of duties. This concept is applicable to physical environments as well as network and host security. Separation of duty ensures that for any given task, more than one individual needs to be involved. The task is broken into different duties, each of which is accomplished by a separate individual. By implementing a task in this manner, no single individual can abuse the system for his or her own gain. This principle has been implemented in the business world, especially financial institutions, for many years. A simple example is a system in which one individual is required to place an order and a separate person is needed to authorize the purchase.

While separation of duties provides a certain level of checks and balances, it is not without its own drawbacks. Chief among these is the cost required to accomplish the task. This cost is manifested in both time and money. More than one individual is required when a single person could accomplish the task, thus potentially increasing the cost of the task. In addition, with more than one individual involved, a certain delay can be expected as the task must proceed through its various steps.

Implicit Deny

What has become the Internet was originally designed as a friendly environment where everybody agreed to abide by the rules implemented in the various protocols. Today, the Internet is no longer the friendly playground of researchers that it once was. This has resulted in different approaches that might at first seem less than friendly but that are required for security purposes. One of these approaches is implicit deny. Frequently in the network world, decisions concerning access must be made. Often a series of rules will be used to determine whether or not to allow access. If a particular situation is not covered by any of the other rules, the implicit deny approach states that access should not be granted. In other words, if no rule would allow access, then access should not be granted. Implicit deny applies to situations involving both authorization and access.

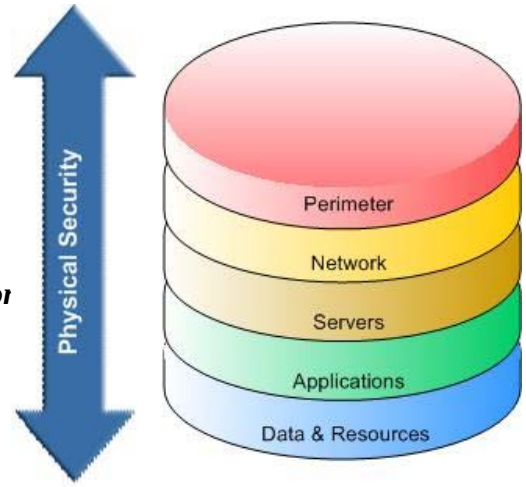
Job Rotation

An interesting approach to enhance security that is gaining increasing attention is through job rotation. The benefits of rotating individuals through various jobs in an organization's IT department have been discussed for a while. By rotating through jobs, individuals gain a better perspective of how the various parts of IT can enhance (or hinder) the business. Since security is often a misunderstood aspect of IT, rotating individuals through security positions can result in a much wider understanding of the security problems throughout the organization. It also can have the side benefit of not relying on any one individual too heavily for security expertise. When all security tasks are the domain of one employee, and if that individual were to leave suddenly, security at the organization could suffer. On the other hand, if security tasks were understood by many different individuals, the loss of any one individual would have less of an impact on the organization.

One significant drawback to job rotation is relying on it too heavily. The IT world is very technical and often expertise in any single aspect takes years to develop. This is especially true in the security environment. In addition, the rapidly changing threat environment with new vulnerabilities and exploits routinely being discovered requires a level of understanding that takes considerable time to acquire and maintain.

Layered Security

A bank does not protect the money that it stores only by placing it in a vault. It uses one or more security guards as a first defense to watch for suspicious activities and to secure the facility when the bank is closed. It probably uses monitoring systems to watch various activities that take place in



the bank, whether involving customers or employees. The vault is usually located in the center of the facility, and layers of rooms or walls also protect access to the vault. Access control ensures that the people who want to enter the vault have been granted the appropriate authorization before they are allowed access, and the systems, including manual switches, are connected directly to the police station in case a determined bank robber successfully penetrates any one of these layers of protection.

Networks should utilize the same type of layered security architecture. No system is 100 percent secure and nothing is foolproof, so no single specific protection mechanism should ever be trusted alone. Every piece of software and every device can be compromised in some way, and every encryption algorithm can be broken by someone with enough time and resources. The goal of security is to make the effort of actually accomplishing a compromise more costly in time and effort than it is worth to a potential attacker.

Consider, for example, the steps an intruder has to take to access critical data held within a company's back-end database. The intruder will first need to penetrate the firewall and use packets and methods that will not be identified and detected by the IDS. The attacker will have to circumvent an internal router performing packet filtering and possibly penetrate another firewall that is used to separate one internal network from another. From here, the intruder must break the access controls on the database, which means performing a dictionary or brute force attack to be able to authenticate to the database software. Once the intruder has gotten this far, he still needs to locate the data within the database. This can in turn be complicated by the use of access control lists (ACLs) outlining who can actually view or modify the data. That's a lot of work.

Diversity of Defense

Diversity of defense is a concept that complements the idea of various layers of security; layers are made dissimilar so that even if an attacker knows how to get through a system making up one layer, she might not know how to get through a different type of layer that employs a different system for security.

If, for example, an environment has two firewalls that form a demilitarized zone (a DMZ is the area between the two firewalls that provides an environment where activities can be more closely monitored), one firewall can be placed at the perimeter of the Internet and the DMZ. This firewall will analyze traffic that passes through that specific access point and enforces certain types of restrictions. The other firewall can be placed between the DMZ and the internal network. When applying the diversity of defense concept, you should set up these two firewalls to filter for different types of traffic and provide different types of restrictions. The first firewall, for example, can make sure that no File Transfer Protocol (FTP), Simple Network Management Protocol (SNMP), or Telnet traffic enters the network, but allow Simple Mail Transfer Protocol (SMTP), Secure Shell (SSH), Hypertext Transfer Protocol (HTTP), and SSL traffic through. The second firewall may not allow SSL or SSH through and can interrogate SMTP and HTTP traffic to make sure that certain types of attacks are not part of that traffic.

Another type of diversity of defense is to use products from different vendors. Every product has its own security vulnerabilities that are usually known to experienced attackers in the community. A Check Point firewall, for example, has different security issues and settings than a Sidewinder firewall; thus, different exploits can be used to crash or compromise them in some fashion. Combining this type of diversity with the preceding example, you might use the Check Point firewall as the first line of defense. If attackers are able to penetrate it, they are less likely to get through the next firewall if it is a Cisco PIX or Sidewinder firewall (or another maker's firewall).

Security Through Obscurity

With security through obscurity, security is considered effective if the environment and protection mechanisms are confusing or supposedly not generally known. Security through obscurity uses the approach of protecting something by hiding it—out of sight, out of mind. Non-computer examples of this concept include hiding your briefcase or purse if you leave it in the car so that it is not in plain view, hiding a house key under a ceramic frog on your porch, or pushing your favorite ice cream to the back of the freezer so that nobody else will see it. This approach, however, does not provide actual protection of the object. Someone can still steal the purse by breaking into the car, lift the ceramic frog and find the key, or dig through the items in the freezer to find the ice cream. Security through obscurity may make someone work a little harder to accomplish a task, but it does not prevent anyone from eventually succeeding.

Similar approaches occur in computer and network security when attempting to hide certain objects. A network administrator can, for instance, move a service from its default port to a different port so that others will not know how to access it as easily, or a firewall can be configured to hide specific information about the internal network in the hope that potential attackers will not obtain the information for use in an attack on the network.

Keep It Simple

The terms security and complexity are often at odds with each other, because the more complex something is, the more difficult it is to understand, and you cannot truly secure something if you do not understand it. Another reason complexity is a problem within security is that it usually allows too many opportunities for something to go wrong. An application with 4000 lines of code has far fewer places for buffer overflows, for example, than an application with 2 million lines of code.

As with any other type of technology, when something goes wrong with security mechanisms, a troubleshooting process is used to identify the problem. If the mechanism is overly complex, identifying the root of the problem can be overwhelming if not impossible. Security is already a very complex issue because many variables are involved, many types of attacks and vulnerabilities are possible, many different types of resources must be secure, and many different ways can be used to secure them. You want your security processes and tools to be as simple and elegant as possible. They should be simple to troubleshoot, simple to use, and simple to administer.

Access Control

The term access control describes a variety of protection schemes. It sometimes refers to all security features used to prevent unauthorized access to a computer system or network. In this sense, it may be confused with authentication. More properly, access is the ability of a subject (such as an individual or a process running on a computer system) to interact with an object (such as a file or hardware device). Authentication, on the other hand, deals with verifying the identity of a subject.

To understand the difference, consider the example of an individual attempting to log in to a computer system or network. Authentication is the process used to verify to the computer system or network that the individual is who he claims to be. The most common method to do this is through the use of a user ID and password. Once the individual has verified his identity, access controls regulate what the individual can actually do on the system—just because a person is granted entry to the system does not mean that he should have access to all data the system contains.

Consider another example. When you go to your bank to make a withdrawal, the teller at the window will verify that you are indeed who you claim to be by asking you to provide some form of identification with your picture on it, such as your driver's license. You might also have to provide your bank account number. Once the teller verifies your identity, you will have proved that you are a valid (authorized) customer of this bank. This does not, however, mean that you have the ability to view all information that the bank protects—such as your neighbor's account balance. The teller will control what information, and funds, you can access and will grant you access only to information for which you are authorized to see. In this example, your identification and bank account number serve as your method of authentication and the teller serves as the access control mechanism.

In computer systems and networks, access controls can be implemented in several ways. An access control matrix provides the simplest framework for illustrating the process. In this matrix, the system is keeping track of two processes, two files, and one hardware device. Process 1 can read both File 1 and File 2 but can write only to File 1. Process 1 cannot access Process 2, but Process 2 can execute Process 1. Both processes have the ability to write to the printer. While simple to understand, the access control matrix is seldom used in computer systems because it is extremely costly in terms of storage space and processing. Imagine the size of an access control matrix for a large network with hundreds of users and thousands of files. The actual mechanics of how access controls are implemented in a system varies, though access control lists (ACLs) are common. An ACL is nothing more than a list that contains the subjects that have access rights to a particular object. The list identifies not only the subject but the specific access granted to the subject for the object. Typical types of access include read, write, and execute as indicated in the example access control matrix.

No matter what specific mechanism is used to implement access controls in a computer system or network, the controls should be based on a specific model of access. Several different models are discussed in security literature, including discretionary access control (DAC), mandatory access control (MAC), role-based access control (RBAC), and rule-based access control (also RBAC).

Discretionary Access Control

Both discretionary access control and mandatory access control are terms originally used by the military to describe two different approaches to controlling an individual's access to a system. As defined by the "Orange Book," a Department of Defense document that at one time was the standard for describing what constituted a trusted computing system, DACs are "a means of restricting access to objects based on the identity of subjects and/or groups to which they belong. The controls are discretionary in the sense that a subject with a certain access permission is capable of passing that permission (perhaps indirectly) on to any other subject." While this might appear to be confusing "government-speak," the principle is rather simple. In systems that employ DACs, the owner of an object can decide which other subjects can have access to the object and what specific access they can have. One common method to accomplish this is the permission bits used in UNIX-based systems. The owner of a file can specify what permissions (read/write/execute) members in the same group can have and also what permissions all others can have. ACLs are also a common mechanism used to implement DAC.

Mandatory Access Control

A less frequently employed system for restricting access is mandatory access control. This system, generally used only in environments in which different levels of security classifications exist, is much more restrictive regarding what a user is allowed to do. Referring to the "Orange Book," a mandatory access control is "a means of restricting access to objects based on the sensitivity (as represented by a label) of the information contained in the objects and the formal authorization (i.e., clearance) of subjects to access information of such sensitivity." In this case, the owner or subject can't determine whether access is to be granted to another subject; it is the job of the operating system to decide.

Role-Based Access Control

ACLs can be cumbersome and can take time to administer properly. Another access control mechanism that has been attracting increased attention is the role-based access control (RBAC). In this scheme, instead of each user being assigned specific access permissions for the objects associated with the computer system or network, each user is assigned a set of roles that he or she may perform. The roles are in turn assigned the access permissions necessary to perform the tasks associated with the role. Users will thus be granted permissions to objects in terms of the specific duties they must perform—not according to a security classification associated with individual objects.

Rule-Based Access Control

The first thing that you might notice is the ambiguity that is introduced with this access control method also using the acronym RBAC. Rule-based access control again uses objects such as ACLs to help determine whether access should be granted or not. In this case, a series of rules are contained in the ACL and the determination of whether to grant access will be made based on these rules. An example of such a rule is one that states that no employee may have access to the payroll file after hours or on weekends.

As with MAC, users are not allowed to change the access rules, and administrators are relied on for this. Rule-based access control can actually be used in addition to or as a method of implementing other access control methods. For example, MAC methods can utilize a rule-based approach for implementation.

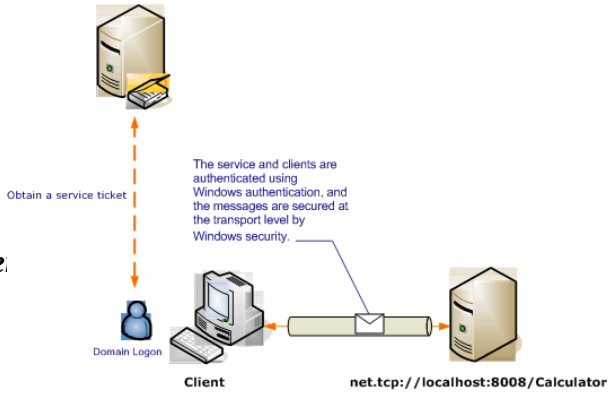
Authentication

Access controls define what actions a user can perform or what objects a user can access. These controls assume that the identity of the user has already been verified. It is the job of authentication mechanisms to ensure that only valid users are admitted. Described another way, authentication uses some mechanism to prove that you are who you claim to be. Three general methods are used in authentication. To verify your identity, you can provide the following:

- Something you know
- Something you have
- Something you are (something unique about you)

The most common authentication mechanism is to provide something that only you, the valid user, should know. The most frequently used example of this is the common user ID (or username) and password. In theory, since you are not supposed to share your password with anybody else, only you should know your password, and thus by providing it you are proving to the system that you are who you claim to be. In theory, this should be a fairly decent method to provide authentication. Unfortunately, for a variety of reasons, such as the fact that people have a tendency to choose very poor and easily guessed passwords, this technique is not as reliable as it should be. Other authentication mechanisms are consequently always being developed and deployed.

Another method to provide authentication involves the use of something that only valid users should have in their possession. A physical-world example of this would be a simple lock and key. Only those individuals with the correct key will be able to open the lock and thus provide admittance to a house, car, office, or whatever the lock was protecting. A similar method can be used to authenticate users for a computer system or network (though the key may be electronic and may reside on a smart card or similar device). The problem with this technology is that people will lose their keys (or cards), which means they can't log in to the system and somebody else who finds



the key can then access the system, even though that person is not authorized. To address this problem, a combination of the something-you-know/something-you-have methods is often used so that the individual with the key can also be required to provide a password or passcode. The key is useless unless you know this code. An example of this is the ATM card most of us carry. The card is associated with a personal identification number (PIN), which only you should know. Knowing the PIN without having the card is useless, just as having the card without knowing the PIN will not give you access to your account.

Operational Organizational Security

To some, the solution to securing an organization's computer systems and network is simply the implementation of various security technologies. Prevention technologies are designed to keep individuals from being able to gain access to systems or data they are not authorized to use. They are intended to prevent unauthorized access. A common prevention technology is the implementation of logical access controls. Although an important element of security, the implementation of any technological solution should be based upon an organizational security policy. In this chapter you will learn about various organizational and operational elements of security. Some of these, such as the establishment of security policies, standards, guidelines, and procedures, are activities that fall in the prevention category of the operational model of computer security.

Others, such as the discussion on social engineering, come under the category of detection. All of these components, no matter which part of the operational model they fall under, need to be combined in a cohesive operational security program for your organization.

Policies, Standards, Guidelines, and Procedures

A security program (the total of all technology, processes, procedures, metrics, training, and personnel that are part of the organization's approach to addressing security) should be based on an organization's security policies, procedures, standards, and guidelines that specify what users and administrators should be doing to maintain the security of the systems and network. Collectively, these documents provide the guidance needed to determine how security will be implemented in the organization. Given this guidance, the specific technology and security mechanisms required can be planned for. Policies are high-level, broad statements of what the organization wants to accomplish. Standards are mandatory elements regarding the implementation of a policy.

Some standards can be externally driven. Government regulations for banking and financial institutions, for example, require that certain security measures be taken. Other standards may be set by the organization to meet its own security goals. Guidelines are

recommendations relating to a policy. The key term in this case is recommendation—guidelines are not mandatory steps. Procedures are the step-by-step instructions on how to implement policies in the organization.

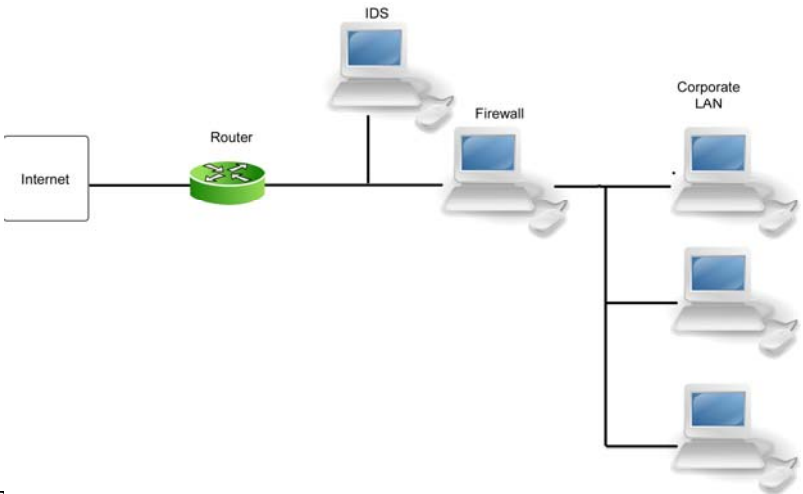
Just as the network itself constantly changes, the policies, standards, guidelines, and procedures should be included in living documents that are periodically evaluated and changed as necessary. The constant monitoring of the network and the periodic review of the relevant documents are part of the process that is the operational model. This operational process consists of four basic steps:

- Plan (adjust) for security
- Implement the plans
- Monitor the implementation
- Evaluate the effectiveness

In the first step, you develop the policies, procedures, and guidelines that will be implemented and design the security components that will protect your network. Once these are designed and developed, you can implement the plans. Next, you monitor to ensure that both the hardware and the software as well as the policies, procedures, and guidelines are working to secure your systems. Finally, you evaluate the effectiveness of the security measures you have in place. The evaluation step can include a vulnerability assessment (an attempt to identify and prioritize the list of vulnerabilities within a system or network) and penetration test (a method to check the security of a system by simulating an attack by a malicious individual) of your system to ensure the security is adequate. After evaluating your security posture, you begin again with step one, this time adjusting the security mechanisms you have in place, and then continue with this cyclical process.

The Security Perimeter

The discussion to this point has not mentioned the specific technology used to enforce operational and organizational security or a description of the various components that constitute the organization's security perimeter. If the average administrator were asked to draw a diagram depicting the various components of her network, the diagram would probably look something like Figure 2-1.



This diagram includes the major components typically found in a network. A connection to the Internet generally has some sort of protection attached to it such as a firewall. An

intrusion detection system (IDS), also often a part of

Figure 2-1

the security perimeter for the organization, can be on the inside of the firewall, or the outside or it may in fact be on both sides. The specific location depends on the company and what it seeks to protect against (that is, insider threats or external threats). Beyond this security perimeter is the corporate LAN. Figure 2-1 is obviously a simple depiction—an actual network can have numerous subnets and extranets—but the basic components are present. Unfortunately, if this were the diagram provided by the administrator to show the organization's basic network structure, the administrator would have missed a very important component. A more astute administrator would provide a diagram more like Figure 2-2.

This diagram includes the other important network found in every organization, the telephone network that is connected to the public switched telephone network (PSTN), otherwise known as the phone company. The organization may or may not have any authorized modems, but the savvy administrator would realize that because the potential exists for unauthorized modems, the telephone network must be included as a possible source of access for the network. When considering the policies, procedures, and guidelines needed to implement security for the organization, both networks need to be considered.

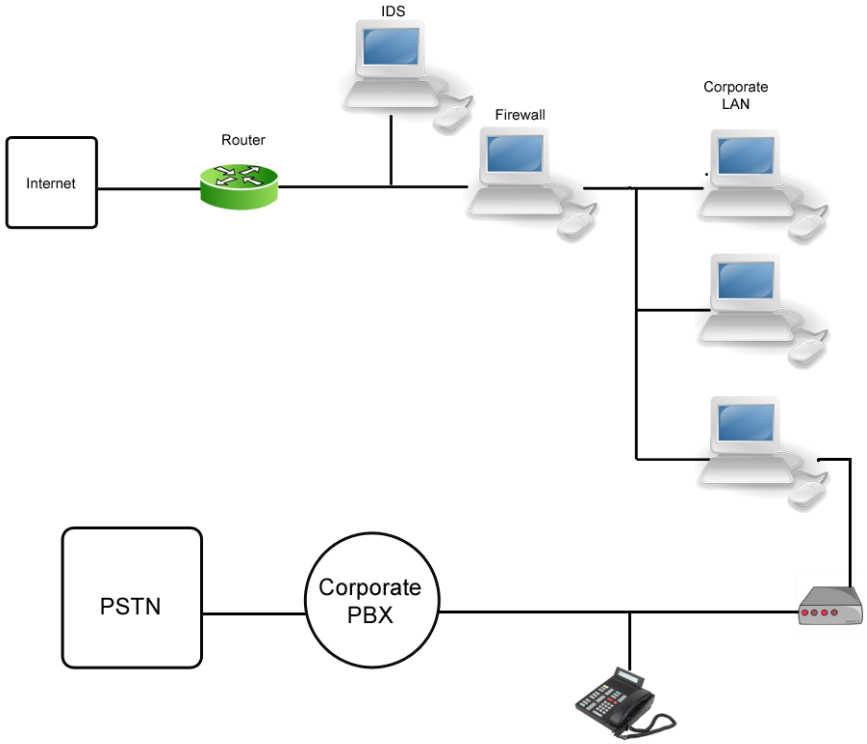


Figure 2.2

While Figure 2-2 provides a more comprehensive view of the various components that need to be protected, it is still incomplete. Most experts will agree that the biggest danger to any organization does not come from external attacks but rather from the insider—a disgruntled employee or somebody else who has physical access to the facility. Given physical access to an office, a knowledgeable attacker will quickly be able to find the information he needs to gain access to the organization's computer systems and network. Consequently, every organization also needs security policies, procedures, and guidelines that cover physical security and every security administrator should be concerned with these as well. While physical security (which can include such things as locks, cameras, guards and entry points, alarm systems, and physical barriers) will probably not fall under

the purview of the security administrator, the operational state of the organization's physical security measures is just as important as many of the other network-centric measures.

Logical Access Controls
Access control lists (ACLs) are as important to logical access controls as they are to the control of physical access to the organization and its resources. An ACL is simply a list of the individuals (or groups) that are granted access to a specific resource. It can also include the type of access they have (that is, what actions they can perform on or with the resource). Logical access controls refer to those mechanisms that are used to control who may gain electronic access (access to data or resources from a computer system or network as opposed to physical access to the system itself) to the organization's computer systems and networks. Before setting the system's access controls, you must establish the security policies that the settings will be based upon.

Access Control Policies
As mentioned, policies are statements of what the organization wants to accomplish. The organization needs to identify goals and intentions for many different aspects of security. Each aspect will have associated policies and procedures.

Group Policy
Operating systems such as Windows and Linux allow administrators to organize users into groups. This is used to create categories of users for which similar access policies can be established. Using groups saves the administrator time, as adding a new user will not require that he create a completely new user profile; instead the administrator would determine to which group the new user belongs and then add the user to that group. Examples of groups commonly found include administrator, user, and guest.

Password Policy
Since passwords are the most common authentication mechanism, it is imperative that organizations have a policy addressing them. The list of authorized users will form the basis of the ACL for the computer system or network that the passwords will help control. The password policy should address the procedures used for selecting user passwords (specifying what is considered an acceptable password in the organization in terms of the character set and length, for example), the frequency with which they must be changed, and how they will be distributed. Procedures for creating new passwords should an employee forget her old password also need to be addressed, as well as the acceptable handling of passwords (for example, they should not be shared with anybody else, they should not be written down, and so on). It might also be useful to have the policy address the issue of password cracking by administrators, in order to discover weak passwords selected by employees.

Domain Password Policy
Domains are logical groups of computers that share a central directory database. The database contains information about the user accounts and security information for all

resources identified within the domain. Each user within the domain is assigned her own unique account (that is, a domain is not a single account shared by multiple users), which is then assigned access to specific resources within the domain. In operating systems that provide domain capabilities, the password policy is set in the root container for the domain and will apply to all users within that domain. Setting a password policy for a domain is similar to setting other password policies in that the same critical elements need to be considered (password length, complexity, life, and so on). If a change to one of these elements is desired for a group of users, a new domain will need to be created. In a Microsoft Windows operating system that employs Active Directory, the domain password policy can be set in the Active Directory Users and Computers menu in the Administrative Tools section of the Control Panel.

Usernames and Passwords
Policies regarding selection of usernames and passwords must weigh usability versus security. At one end of the spectrum is usability, which would dictate that the username be simple and easy to remember, such as the user's first and last name separated by a period or the user's first initial followed by the last name. This makes it easy for the user to remember the user (account) name and makes it easy for other individuals to remember a user's username (since the username and e-mail name are generally similar).

At the same time, however, adhering to a simple policy such as this also makes it easy for a potential attacker to guess a valid account name, which can then be used in an attempt to guess a username/password combination. At the other end of the spectrum is the generation of a completely random series of characters (such as xzf258) to be assigned to a user for a username. Aliases can be used for e-mail so that the more common first name/last name format can still be used for communication with users. The advantage of this random assignment is that it will be more difficult for an attacker to guess a valid username; however, it has the disadvantage of being difficult for the user to remember.

Time of Day Restrictions
Some systems allow for the specification of time of day restrictions in their access control policies. This means that a user's access to the system or specific resources can be restricted to certain times of the day and days of the week. If a user normally accesses certain resources during normal business hours, an attempt to access these resources outside this time period (either at night or on the weekend) might indicate an attacker has gained access to the account. Specifying time of day restrictions can also serve as a mechanism to enforce internal controls of critical or sensitive resources. Obviously, a drawback to enforcing time of day restrictions is that it means that a user can't go to work outside of normal hours in order to "catch up" with work tasks. As with all security policies, usability and security must be balanced in this policy decision.

Account and Password Expiration
Another common restriction that can be enforced in many access control mechanisms is either (or both) an account expiration or password expiration feature. This allows administrators to specify a period of time for which a password or an account will be

active. For password expiration, when the expiration date is reached, the user will generally be asked to create a new password. This means that if the password (and thus the account) has been compromised when the expiration date is reached and a new password is set, the attacker will again (hopefully) be locked out of the system. The attacker can't change the password himself since the user would then be locked out and would contact an administrator to have the password reset, thus again locking out the attacker.

The attacker could set a new password, and then attempt to reset it to the original password. This would mean that a new expiration time would be set for the account but would keep the same password and would not lock the user out. This is one reason why a password history mechanism should be used. The history is used to keep track of previously used passwords so that they cannot be reused. An account expiration is similar, except that it is generally put in place because a specific account is intended for a specific purpose of limited duration. When an account has expired, it cannot be used unless the expiration deadline is extended.

File and Print Resources

The desire for a collaborative work environment often results in file sharing on servers. In a similar manner, print resources are also often shared so that many users can access high-cost resources. In the past, the potential for security problems associated with shared resources (it was often difficult to isolate who could or could not use the resource if it was opened for sharing) had led to some security administrators simply prohibiting sharing. With some of the more current operating systems, however, sharing can be accomplished with a reasonable balance between it and security. Strict policies regarding sharing need to be established. Some files should not be shared (such as a user's profile folder, for example), so allowing for a blanket sharing of files between users should be avoided. Instead, specific files within folders should be designated and managed through group policies. Similar care should be taken when deciding what print resources should be shared.

Logical Tokens

A token is an object that a user must have and present to the system to gain access to some resource or the system itself. Special hardware devices can be used as tokens that need to be inserted into the machine or a special reader, or that can provide some information (such as a one-time code) that must be supplied to the system to obtain access. A problem with all of these methods is that they require that the user have the physical device on hand to gain access. If the user loses the token or forgets it, she will be unable to access the resource.

Social Engineering

Social engineering is the process of convincing an authorized individual to provide confidential information or access to an unauthorized individual. Social engineering takes advantage of what continually turns out to be the weakest point in our security perimeter— the humans. Kevin Mitnick, a convicted cybercriminal turned security

consultant, once stated, “Don’t rely on network safeguards and firewalls to protect your information. Look to your most vulnerable spot. You’ll usually find that vulnerability lies in your people.” In 2000, after being released from jail, Mitnick testified before Congress and spoke on several other occasions about social engineering and how effective it is. He stated that he “rarely had to resort to a technical attack” because of how easily information and access could be obtained through social engineering.

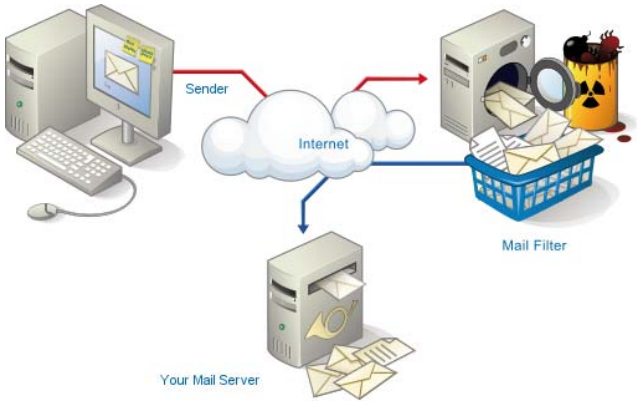
The goal of social engineering is to gradually obtain the pieces of information necessary to make it to the next step. This is done repeatedly until the ultimate goal is reached. If social engineering is such an effective means of gaining unauthorized access to data and information, how can it be stopped? The most effective means is through the training and education of users, administrators, and security personnel. All employees should be instructed in the techniques that attackers might use and trained to recognize when a social engineering attack is being attempted. One important aspect of this training is for employees to recognize the type of information that should be protected and also how seemingly unimportant information can be combined with other pieces of information to potentially divulge sensitive information. This is known as data aggregation.

In addition to the direct approach to social engineering, attackers can use other indirect means to obtain the information they are seeking. These include phishing, vishing, shoulder surfing, and dumpster diving and are discussed in the following sections. Again, the first defense against any of these methods to gather information to be used in later attacks is a strong user education and awareness training program.

Phishing

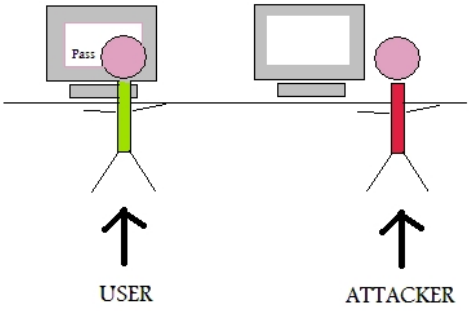
Phishing (pronounced “fishing”) is a type of social engineering in which an individual attempts to obtain sensitive information from a user by masquerading as a trusted entity in an e-mail or instant message sent to the user. The type of information that the attacker attempts to obtain include usernames, passwords, credit card numbers, or details on the user’s bank account. The message sent often encourages the user to go to a web site that appears to be for a reputable entity such as PayPal or eBay, both of which have frequently been used in phishing attempts. The web site the user actually visits will not be owned by the reputable organization, however, and will ask the user to supply information that can be used in a later attack. Often the message sent to the user will tell a story about the user’s account having been compromised, and for security purposes they are encouraged to enter their account information to verify the details.

Vishing



Vishing is a variation of phishing that uses voice communication technology to obtain the information the attacker is seeking. Vishing takes advantage of the trust that most people place in the telephone network. Users are unaware that attackers can spoof calls from legitimate entities using voice over IP (VoIP) technology. Voice messaging can also be compromised and used in these attempts. Generally, the attackers are hoping to obtain credit card numbers or other information that can be used in identity theft. The user may receive an e-mail asking him to call a number that is answered by a potentially compromised voice message system. Users may also receive a recorded message that appears to come from a legitimate entity. In both cases, the user will be encouraged to respond quickly and provide the sensitive information so that access to an account is not blocked. If a user ever receives a message that claims to be from a reputable entity and is asking for sensitive information, he should not provide it but instead use the
Internet or examine a legitimate account statement to find a phone number that can be used to contact the entity. The user can then verify that the message received was legitimate or report the vishing attempt.

Shoulder Surfing



Shoulder surfing does not involve direct contact with the user, but instead involves the attacker directly observing the target entering sensitive information on a form, keypad, or keyboard. The attacker may simply look over the shoulder of the user at work or the attacker can set up a camera or use binoculars to view users entering sensitive data. The attacker can attempt to obtain information such as a PIN at an automated teller machine, an access control entry code at a secure gate or door, or calling card or credit card numbers. Some locations now use a small shield to surround a keypad so that it is difficult to observe somebody entering information. More sophisticated systems can actually scramble the location of the numbers so that the top row at one time includes the numbers 1, 2, and 3 and the next time 4, 8, and 0. While this makes it a bit slower for the user to enter information, it does mean that a person attempting to observe what numbers are pressed will not be able to press the same buttons/pattern since the location of the numbers have changed.

Dumpster Diving

Dumpster diving is not a uniquely computer security–related activity. It refers to the activity of sifting through an individual's or organization's trash for things that the dumpster diver might find valuable. In the non-security realm, this can be anything from empty aluminum cans to articles of clothing or discarded household items. From a computer security standpoint, the diver is looking for information that can be obtained from listings or printouts, manuals, receipts, or even yellow sticky notes. The information can include credit card or bank account numbers, user IDs or passwords, details about the type of software or hardware platforms that are being used, or even company sensitive information. In most locations, trash is no longer considered private property after it has been discarded (and even where dumpster diving is illegal, little enforcement occurs). An organization should have policies about discarding materials. Sensitive information should be shredded and the organization should consider securing the trash receptacle so that individuals can't forage through it. People should also consider shredding personal or sensitive information that they wish to discard in their own trash. A reasonable quality shredder is inexpensive and well worth the price when compared with the potential loss that could occur as a result of identity theft.

Hoaxes

At first glance, it might seem that a hoax related to security would be considered a nuisance and not a real security issue. This might be the case for some hoaxes, especially those of the urban legend type, but the reality of the situation is that a hoax can be very damaging if it causes users to take some sort of action that weakens security. One real hoax, for example, told the story of a new, highly destructive piece of malicious software. It instructed users to check for the existence of a certain file and to delete it if the file was found. In reality, the file mentioned was an important file that was used by the operating system, and deleting it caused problems the next time the system was booted. The damage caused by users modifying security settings can be serious. As with other forms of social engineering, training and awareness are the best and first line of defense for users. Users should be trained to be suspicious of unusual e-mails and stories and should know who to contact in the organization to verify the validity if they are received.

Organizational Policies and Procedures

Policies are high-level statements created by management that lay out the organization's positions on particular issues. Policies are mandatory but are not specific in their details.

Policies are focused on the result, not the methods for achieving that result. Procedures are generally step-by-step instructions that prescribe exactly how employees are expected to act in a given situation or to accomplish a specific task. Although standard policies can be described in general terms that will be applicable to all organizations, standards and procedures are often organization-specific and driven by specific organizational policies.

Regarding security, every organization should have several common policies in place in addition to those already discussed relative to access control methods. These policies include acceptable use policies, due care, separation of duties, and policies governing the protection of personally identifiable information (PII), and they are addressed in the following sections. Other important policy-related issues covered here include privacy, service level agreements, human resources policies, codes of ethics, and policies governing incident response.

Security Policies
In keeping with the high-level nature of policies, the security policy is a high-level statement produced by senior management that outlines what security means to the organization and the organization's goals for security. The main security policy can then be broken down into additional policies that cover specific topics. Statements such as "this organization will exercise the principle of least access in its handling of client information" would be an example of a security policy. The security policy can also describe how security is to be handled from an organizational point of view (such as describing which office and corporate officer or manager oversees the organization's security program).

Change Management
The purpose of change management is to ensure proper procedures are followed when modifications to the IT infrastructure are made. These modifications can be prompted by a number of different reasons including new legislation, updated versions of software or hardware, implementation of new software or hardware, or improvements to the infrastructure. The term "management" implies that this process should be controlled in some systematic way, and that is indeed the purpose. Changes to the infrastructure can have a detrimental impact on operations. New versions of operating systems or application software can be incompatible with other software or hardware the organization is using. Without a process to manage the change, an organization can suddenly find itself unable to conduct business.

Classification of Information
A key component of IT security is the protection of the information processed and stored on the computer systems and network. Organizations deal with many different types of information, and they need to recognize that not all information is of equal importance or sensitivity. This prompts a classification of information into various categories, each with its own requirements for its handling. Factors that affect the classification of specific information include its value to the organization (what will be the impact to the organization if it loses this information?), its age, and laws or regulations that govern its

protection. The most widely known classification of information is that implemented by the government and military, which classifies information into categories such as confidential, secret, and top secret. Businesses have similar desires to protect information but can use categories such as publicly releasable, proprietary, company confidential or for internal use only. Each policy for a classification of information should describe how it should be protected, who may have access to it, who has the authority to release it and how it should be destroyed. All employees of the organization should be trained in the procedures for handling the information that they are authorized to access. Discretionary and mandatory access control techniques use classifications as a method to identify who may have access to what resources.

Acceptable Use
An acceptable use policy (AUP) outlines what the organization considers to be the appropriate use of company resources, such as computer systems, e-mail, Internet, and networks. Organizations should be concerned with the personal uses of organizational assets that do not benefit the company.

The goal of the policy is to ensure employee productivity while limiting organizational liability through inappropriate use of the organization's assets. The policy should clearly delineate what activities are not allowed. Issues such as the use of resources to conduct personal business, installation of hardware or software, remote access to systems and networks, the copying of company-owned software, and the responsibility of users to protect company assets, including data, software, and hardware should be addressed. Statements regarding possible penalties for ignoring any of the policies (such as termination) should also be included.

Internet Usage Policy In today's highly connected environment, employee use of access to the Internet is of particular concern. The goal for the Internet usage policy is to ensure maximum employee productivity and to limit potential liability to the organization from inappropriate use of the Internet in a workplace. The Internet provides a tremendous temptation for employees to waste hours as they surf the Web for the scores of the important games from the previous night, conduct quick online stock transactions, or read the review of the latest blockbuster movie everyone is talking about. Obviously, every minute they spend conducting this sort of activity is time they are not productively engaged in the organization's business and their jobs. In addition, allowing employees to visit sites that may be considered offensive to others (such as pornographic or hate sites) can open the company to accusations of condoning a hostile work environment and result in legal liability.

E-Mail Usage Policy Related to the Internet usage policy is the e-mail usage policy, which deals with what the company will allow employees to send in terms of e-mail. This policy should spell out whether non-work e-mail traffic is allowed at all or is at least severely restricted. It needs to cover the type of message that would be considered inappropriate to send to other employees (for example, no offensive language, no sex related or ethnic jokes, no harassment, and so on). The policy should also specify any

disclaimers that must be attached to an employee's message sent to an individual outside the company.

Due Care and Due Diligence
Due care and due diligence are terms used in the legal and business community to address issues where one party's actions might have caused loss or injury to another's. Basically, the law recognizes the responsibility of an individual or organization to act reasonably relative to another with diligence being the degree of care and caution exercised. Reasonable precautions need to be taken that indicate that the organization is being responsible. In terms of security, it is expected that organizations will take reasonable precautions to protect the information that it maintains on other individuals. Should a person suffer a loss as a result of negligence on the part of an organization in terms of its security, a legal suit can be brought against the organization.

Due Process
Due process is concerned with guaranteeing fundamental fairness, justice, and liberty in relation to an individual's legal rights. In the United States, due process is concerned with the guarantee of an individual's rights as outlined by the Constitution and Bill of Rights. Procedural due process is based on the concept of what is "fair." Also of interest is the recognition by courts of a series of rights that are not explicitly specified by the Constitution but that the courts have decided are implicit in the concepts embodied by the Constitution. An example of this is an individual's right to privacy. From an organization's point of view, due process may come into play during an administrative action that adversely affects an employee. Before an employee is terminated, for example, were all of the employee's rights protected? An actual example pertains to the rights of privacy regarding employees' e-mail messages. As the number of cases involving employers examining employee e-mails grows, case law is established and the courts eventually settle on what rights an employee can expect. The best thing an employer can do if faced with this sort of situation is to work closely with HR staff to ensure that appropriate policies are followed and that those policies are in keeping with current laws and regulations.

Separation of Duties
Separation of duties is a principle employed in many organizations to ensure that no single individual has the ability to conduct transactions alone. This means that the level of trust in any one individual is lessened, and the ability for any individual to cause catastrophic damage to the organization is also lessened. An example might be an organization in which one person has the ability to order equipment, but another individual makes the payment. An individual who wants to make an unauthorized purchase for his own personal gain would have to convince another person to go along with the transaction.

Need to Know and Least Privilege
Two other common security principles are that of need to know and least privilege. The guiding factor here is that each individual in the organization is supplied with only the

absolute minimum amount of information and privileges she needs to perform her work tasks. To obtain access to any piece of information, the individual must have a justified need to know. In addition, she will be granted only the bare minimum number of privileges that are needed to perform her job.
A policy spelling out these two principles as guiding philosophies for the organization should be created. The policy should also address who in the organization can grant access to information or may assign privileges to employees.

Disposal and Destruction
Many potential intruders have learned the value of dumpster diving. Not only should an organization be concerned with paper trash and discarded objects, but it must also be concerned with the information stored on discarded objects such as computers. Several government organizations have been embarrassed when old computers sold to salvagers proved to contain sensitive documents on their hard drives. It is critical for every organization to have a strong disposal and destruction policy and related procedures.

Privacy

Customers place an enormous amount of trust in organizations to which they provide personal information. These customers expect their information to be kept secure so that unauthorized individuals will not gain access to it and so that authorized users will not use the information in unintended ways. Organizations should have a privacy policy that explains what their guiding principles will be in guarding personal data to which they are given access. In many locations, customers have a legal right to expect that their information is kept private, and organizations that violate this trust may find themselves involved in a lawsuit. In certain sectors, such as health care, federal regulations have been created that prescribe stringent security controls on private information.

Service Level Agreements
Service level agreements (SLAs) are contractual agreements between entities describing specified levels of service that the servicing entity agrees to guarantee for the customer. These agreements clearly lay out expectations in terms of the service provided and support expected, and they also generally include penalties should the described level of service or support not be provided. An organization contracting with a service provider should remember to include in the agreement a section describing the service provider's responsibility in terms of business continuity and disaster recovery. The provider's backup plans and processes for restoring lost data should also be clearly described.

Human Resources Policies
It has been said that the weakest links in the security chain are the humans. Consequently, it is important for organizations to have policies in place relative to its employees.

Policies that relate to the hiring of individuals are primarily important. The organization needs to make sure that it hires individuals that can be trusted with the organization's data and that of its clients. Once employees are hired, they should be kept from slipping into the category of "disgruntled employee." Finally, policies must be developed to address the inevitable point in the future when an employee leaves the organization—either on his own or with the "encouragement" of the organization itself. Security issues must be considered at each of these points.

Code of Ethics

Numerous professional organizations have established codes of ethics for their members. Each of these describes the expected behavior of their members from a high-level standpoint. Organizations can adopt this idea as well. For organizations, a code of ethics can set the tone for how employees will be expected to act and to conduct business. The code should demand honesty from employees and should require that they perform all activities in a professional manner. The code could also address principles of privacy and confidentiality and state how employees should treat client and organizational data. Conflicts of interest can often cause problems, so this could also be covered in the code of ethics.

Cryptography and Applications

Cryptography

Cryptography is the science of encrypting, or hiding, information—something people have sought to do since they began using language. Although language allowed them to communicate with one another, people in power attempted to hide information by controlling who was taught to read and write. Eventually, more complicated methods of concealing information by shifting letters around to make the text unreadable were developed.

The Romans typically used a different method known as a shift cipher. In this case, one letter of the alphabet is shifted a set number of places in the alphabet for another letter. A common modern-day example of this is the ROT13 cipher, in which every letter is rotated 13 positions in the alphabet: n is written instead of a, o instead of b, and so on. These ciphers were simple to use and also simple to break. Because hiding information was still important, more advanced transposition and substitution ciphers were required.

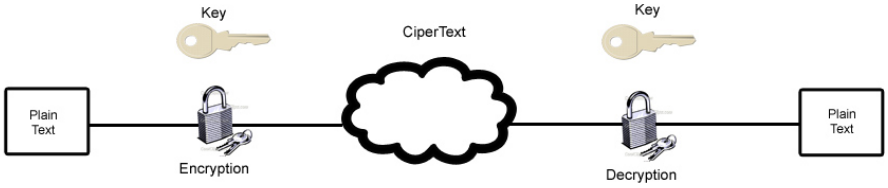
As systems and technology became more complex, ciphers were frequently automated by some mechanical or electromechanical device. A famous example of a modern encryption machine is the German Enigma machine from World War II. This machine used a complex series of substitutions to perform encryption, and interestingly enough it gave rise to extensive research in computers.

Cryptanalysis, the process of analyzing available information in an attempt to return the encrypted message to its original form, required advances in computer technology for complex encryption methods. The birth of the computer made it possible to easily execute the calculations required by more complex encryption algorithms. Today, the computer almost exclusively powers how encryption is performed. Computer technology has also aided cryptanalysis, allowing new methods to be developed, such as linear and differential cryptanalysis. Differential cryptanalysis is done by comparing the input plaintext to the output ciphertext to try and determine the key used to encrypt the information. Linear cryptanalysis is similar in that it uses both plaintext and ciphertext, but it puts the plaintext through a simplified cipher to try and deduce what the key is likely to be in the full version of the cipher.

Algorithms

Every current encryption scheme is based upon an algorithm, a step-by-step, recursive computational procedure for solving a problem in a finite number of steps. The cryptographic algorithm—what is commonly called the encryption algorithm or cipher—is made up of mathematical steps for encrypting and decrypting information. Figure 2.3 shows a diagram of the encryption and decryption process and its parts.

The best algorithms are always public algorithms that have been published for peerreview by other cryptographic and mathematical experts. Publication is important, as any flaws in the system can be revealed by others before actual use of the system. Several proprietary algorithms have been reverse-engineered, exposing the confidential data the algorithms try to protect. Examples of this include the decryption of Nikon's proprietary RAW format white balance encryption, and the cracking of the Exxon Mobil SpeedPass RFID encryption. The use of a proprietary system can actually be less secure than using a published system. While proprietary systems are not made available to be tested by potential crackers, public systems are made public for precisely this purpose.



A system that maintains its security after public testing can be reasonably trusted to be secure. A public algorithm can be more secure because good systems rely on the encryption key to provide security, not the algorithm itself. The actual steps for encrypting data can be published, because without the key, the protected information cannot be accessed. A key is a special piece of data used in both the encryption and decryption processes. The algorithms stay the same in every

implementation, but a different key is used for each, which ensures that even if someone knows the algorithm you use to protect your data, he cannot break your security. A classic example of this is the early shift cipher, known as Caesar's cipher.

Hashing

Hashing functions are commonly used encryption methods. A hashing function is a special mathematical function that performs one-way encryption, which means that once the algorithm is processed, there is no feasible way to use the ciphertext to retrieve the plaintext that was used to generate it. Also, ideally, there is no feasible way to generate two different plaintexts that compute to the same hash value. Figure 3.2 shows a generic hashing process.

Common uses of hashing functions are storing computer passwords and ensuring message integrity. The idea is that hashing can produce a unique value that corresponds to the data entered, but the hash value is also reproducible by anyone else running the developed in 1993, was designed as the algorithm to be used for secure hashing in the U.S. Digital Signature Standard (DSS). It is modeled on the MD4 algorithm and implements fixes in that algorithm discovered by the NSA. It creates message digests 160 bits long that can be used by the Digital Signature Algorithm (DSA), which can then compute the signature of the message. This is computationally simpler, as the message digest is typically much smaller than the actual message—smaller message, less work.

SHA-1 works, as do all hashing functions, by applying a compression function to the data input. It accepts an input of up to 264 bits or less and then compresses down to a hash of 160 bits. SHA-1 works in block mode, separating the data into words first, and then grouping the words into blocks. The words are 32-bit strings converted to hex; grouped together as 16 words, they make up a 512-bit block. If the data that is input to SHA-1 is not a multiple of 512, the message is padded with zeros and an integer describing the original length of the message.

At one time, SHA-1 was one of the more secure hash functions, but it has been found vulnerable to a collision attack. Thus, most people are suggesting that implementations of SHA-1 be moved to one of the other SHA versions. These longer versions, SHA-256, SHA-384, and SHA-512, all have longer hash results, making them more difficult to attack successfully. The added security and resistance to attack in SHA-1 does require more processing power to compute the hash.

SHA-256

SHA-256 is similar to SHA-1, in that it will also accept input of less than 264 bits and reduces that input to a hash. This algorithm reduces to 256 bits instead of SHA-1's 160. Defined in FIPS 180-2 in 2002, SHA-256 is listed as an update to the original FIPS 180 that defined SHA. Similar to SHA-1, SHA-256 will accept 264 bits of input and uses 32-bit words and 512-bit blocks. Padding is added until the entire message is a multiple of 512. SHA-256 uses sixty-four 32-bit words, eight working variables, and results in a hash value of eight 32-bit words, hence 256 bits. SHA-256 is more secure than SHA-1, but the

attack basis for SHA-1 can produce collisions in SHA-256 as well since they are similar algorithms. The SHA standard does have two longer versions, however.

SHA-384
SHA-384 is also similar to SHA-1, but it handles larger sets of data. SHA-384 will accept 2128 bits of input, which it pads until it has several blocks of data at 1024-bit blocks. SHA-384 also used 64-bit words instead of SHA-1's 32-bit words. It uses six 64-bit words to produce the 284-bit hash value.

SHA-512
SHA-512 is structurally similar to SHA-384. It will accept the same 2128 input and uses the same 64-bit word size and 1024-bit block size. SHA-512 does differ from SHA-384 in that it uses eight 64-bit words for the final hash, resulting in 512 bits.

Message Digest
Message Digest (MD) is the generic version of one of several algorithms that are designed to create a message digest or hash from data input into the algorithm. MD algorithms work in the same manner as SHA in that they use a secure method to compress the file and generate a computed output of a specified number of bits. They were all developed by Ronald L. Rivest of MIT.

MD2
MD2 was developed in 1989 and is in some ways an early version of the later MD5 algorithm. It takes a data input of any length and produces a hash output of 128 bits. It is different from MD4 and MD5 in that MD2 is optimized for 8-bit machines, whereas the other two are optimized for 32 bit machines. As with SHA, the input data is padded to become a multiple—in this case a multiple of 16 bytes. After padding, a 16-byte checksum is appended to the message. The message is then processed in 16-byte blocks.

MD4
MD4 was developed in 1990 and is optimized for 32-bit computers. It is a fast algorithm, but it can be subject to more attacks than more secure algorithms like MD5. Like MD2, it takes a data input of some length and outputs a digest of 128 bits. The message is padded to become a multiple of 512, which is then concatenated with the representation of the message's original length.
As with SHA, the message is then divided into blocks and also into 16 words of 32 bits. All blocks of the message are processed in three distinct rounds. The digest is then computed using a four-word buffer. The final four words remaining after compression are the 128-bit hash.
An extended version of MD4 computes the message in parallel and produces two 128-bit outputs—effectively a 256-bit hash. Even though a longer hash is produced, security has not been improved because of basic flaws in the algorithm. Cryptographer Hans Dobbertin has shown how collisions in MD4 can be found in under a minute using just a PC. This vulnerability to collisions applies to 128-bit MD4 as well as 256-bit MD4. Most people are moving away from MD4 to MD5 or a robust version of SHA.

MD5
MD5 was developed in 1991 and is structured after MD4 but with additional security to overcome the problems in MD4. Therefore, it is very similar to the MD4 algorithm, only slightly slower and more secure. MD5 creates a 128-bit hash of a message of any length. Like MD4, it segments the message into 512-bit blocks and then into sixteen 32-bit words. First, the original message is padded to be 64 bits short of a multiple of 512 bits. Then a 64-bit representation of the original length of the message is added to the padded value to bring the entire message up to a 512-bit multiple.

Symmetric Encryption
Symmetric encryption is the older and simpler method of encrypting information. The basis of symmetric encryption is that both the sender and the receiver of the message have previously obtained the same key. This is, in fact, the basis for even the oldest ciphers—the Spartans needed the exact same size cylinder, making the cylinder the "key" to the message, and in shift ciphers both parties need to know the direction and amount of shift being performed. All symmetric algorithms are based upon this shared secret principle, including the unbreakable one-time pad method.

DES
DES, the Data Encryption Standard, was developed in response to the National Bureau of Standards (NBS), now known as the National Institute of Standards and Technology (NIST), issuing a request for proposals for a standard cryptographic algorithm in 1973. NBS received a promising response in an algorithm called Lucifer, originally developed by IBM. The NBS and the NSA worked together to analyze the algorithm's security, and eventually DES was adopted as a federal standard in 1976.

NBS specified that the DES standard had to be recertified every five years. While DES passed without a hitch in 1983, the NSA said it would not recertify it in 1987. However, since no alternative was available for many businesses, many complaints ensued, and the NSA and NBS were forced to recertify it. The algorithm was then recertified in 1993. NIST has now certified the Advanced Encryption Standard (AES) to replace DES.

DES is what is known as a block cipher; it segments the input data into blocks of a specified size, typically padding the last block to make it a multiple of the block size required. In the case of DES, the block size is 64 bits, which means DES takes a 64-bit input and outputs 64 bits of ciphertext. This process is repeated for all 64-bit blocks in the message. DES uses a key length of 56 bits, and all security rests within the key. The same algorithm and key are used for both encryption and decryption.

3DES
Triple DES (3DES) is a variant of DES. Depending on the specific variant, it uses either two or three keys instead of the single key that DES uses. It also spins through the DES algorithm three times via what's called multiple encryption. Multiple encryption can be performed in several different ways. The simplest method of multiple encryption is just to stack algorithms on top of each other—taking plaintext, encrypting it with DES, then

encrypting the first ciphertext with a different key, and then encrypting the second ciphertext with a third key. In reality, this technique is less effective than the technique that 3DES uses, which is to encrypt with one key, then decrypt with a second, and then encrypt with a third.

AES
Because of the advancement of technology and the progress being made in quickly retrieving
DES keys, NIST put out a request for proposals for a new Advanced Encryption Standard (AES). It called for a block cipher using symmetric key cryptography and supporting key sizes of 128, 192, and 256 bits. After evaluation, the NIST had five finalists:

- **MARS** IBM
- **RC6** RSA
- **Rijndael** John Daemen and Vincent Rijmen
- **Serpent** Ross Anderson, Eli Biham, and Lars Knudsen
- **Twofish** Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, and Niels Ferguson

In the fall of 2000, NIST picked Rijndael to be the new AES. It was chosen for its overall security as well as its good performance on limited capacity devices. Rijndael's design was influenced by Square, also written by John Daemen and Vincent Rijmen. Like Square, Rijndael is a block cipher separating data input in 128-bit blocks. Rijndael can also be configured to use blocks of 192 or 256 bits, but AES has standardized on 128-bit blocks. AES can have key sizes of 128, 192, and 256 bits, with the size of the key affecting the number of rounds used in the algorithm. Like DES, AES works in three steps on every block of input data:

1. Add round key, performing an XOR of the block with a subkey.
2. Perform the number of normal rounds required by the key length.
3. Perform a regular round without the mix-column step found in the normal round.

After these steps have been performed, a 128-bit block of plaintext produes a 128-bit block of ciphertext. As mentioned in step 2, AES performs multiple rounds. This is determined by the key size. A key size of 128 bits requires 9 rounds, 192-bit keys will re quire 11 rounds, and 256-bit keys use 13 rounds. Four steps are performed in every round:

1. **Byte sub**. Each byte is replaced by its S-box substitute.
2. **Shift row**. Bytes are arranged in a rectangle and shifted.
3. **Mix column**. Matrix multiplication is performed based upon the arranged rectangle.
4. **Add round key**. This round's subkey is cored in.

These steps are performed until the final round has been completed, and when the final step has been performed, the ciphertext is output.

CAST
CAST is an encryption algorithm similar to DES in its structure. It was designed by Carlisle Adams and Stafford Tavares. CAST uses a 64-bit block size for 64- and 128-bit key versions, and a 128-bit block size for the 256-bit key version. Like DES, it divides the plaintext block into a left half and a right half. The right half is then put through function f and then is XORed with the left half. This value becomes the new right half, and the original right half becomes the new left half. This is repeated for eight rounds for a 64-bit key, and the left and right output is concatenated to form the ciphertext block.

RC
RC is a general term for several ciphers all designed by Ron Rivest—RC officially stands for Rivest Cipher. RC1, RC2, RC3, RC4, RC5, and RC6 are all ciphers in the series. RC1 and RC3 never made it to release, but RC2, RC4, RC5, and RC6 are all working algorithms.

RC2
RC2 was designed as a DES replacement, and it is a variable-key-size block-mode cipher. The key size can be from 8 bits to 1024 bits with the block size being fixed at 64 bits. RC2 breaks up the input blocks into four 16-bit words and then puts them through 18 rounds of one of two operations. The two operations are mix and mash. The sequence in which the algorithms works is as follows:

1. Initialize the input block to words R0 through R3.
2. Expand the key into K0 through K63.
3. Initialize j = 0.
4. Five mix rounds.
5. One mash round.
6. Six mix rounds.
7. One mash round.
8. Five mix rounds.

RC5
RC5 is a block cipher, written in 1994. It has multiple variable elements, numbers of rounds, key sizes, and block sizes. The algorithm starts by separating the input block into two words, A and B.

A = A + S0
B = B + S1
For i = 1 to r
A = ((A XOR B) <<< B) + S2i
B = ((B XOR A) <<< A) + S2i+1

A and B represent the ciphertext output. This algorithm is relatively new, but if configured to run enough rounds, RC5 seems to provide adequate security for current bruteforcing technology. Rivest recommends using at least 12 rounds. With 12 rounds in the algorithm, cryptanalysis in a linear fashion proves less effective than brute-force against RC5, and differential analysis fails for 15 or more rounds. A newer algorithm is RC6.

RC6

RC6 is based on the design of RC5. It uses a 128-bit block size, separated into four words of 32 bits each. It uses a round count of 20 to provide security, and it has three possible key sizes: 128, 192, and 256 bits. The four words are named A, B, C, and D, and the algorithm works like this:

B = B + S0
D = D + S1
For i = 1 – 20
[t = (B * (2B + 1)) <<< 5
u = (D * (2D + 1)) <<< 5
A = ((A XOR t) <<< u) + S2i
C = ((C XOR u) <<< t) + S2i+1
(A, B, C, D) = (B, C, D, A)]
A = A + S42
C = C + S43

The output of A, B, C, and D after 20 rounds is the ciphertext. RC6 is a modern algorithm that runs well on 32-bit computers. With a sufficient number of rounds, the algorithm makes both linear and differential cryptanalysis infeasible. The available key lengths make brute-force attacks extremely time-consuming. RC6 should provide adequate security for some time to come.

RC4

RC4 was created before RC5 and RC6, but it differs in operation. RC4 is a stream cipher, whereas all the symmetric ciphers we have looked at so far have been block-mode ciphers. A stream-mode cipher works by enciphering the plaintext in a stream, usually bit by bit. This makes stream ciphers faster than block-mode ciphers. Stream ciphers accomplish this by performing a bitwise XOR with the plaintext stream and a generated keystream.

Blowfish

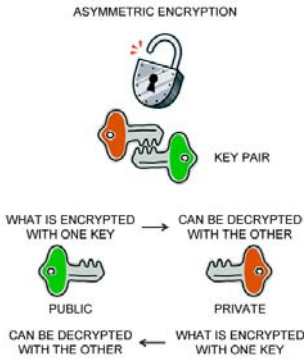
Blowfish was designed in 1994 by Bruce Schneier. It is a block-mode cipher using 64- bit blocks and a variable key length from 32 to 448 bits. It was designed to run quickly on 32-bit microprocessors and is optimized for situations with few key changes. Encryption is done by separating the 64-bit input block into two 32-bit words, and then a function is executed every round. Blowfish has 16 rounds.

IDEA
IDEA (International Data Encryption Algorithm) started out as PES, or Proposed Encryption Cipher, in 1990, and it was modified to improve its resistance to differential cryptanalysis and its name was changed to IDEA in 1992. It is a block-mode cipher using a 64-bit block size and a 128-bit key. The input plaintext is split into four 16-bit segments, A, B, C, and D.

This algorithm is fairly new, but all current cryptanalysis on full, eight-round IDEA shows that the most efficient attack would be to brute-force the key. The 128-bit key would prevent this attack being accomplished, given current computer technology. The only known issue is that IDEA is susceptible to a weak key—a key that is made of all 0s. This weak key is easy to check for, and the weakness is simple to mitigate.

Symmetric Encryption Summary
Symmetric algorithms are important because they are comparatively fast and have few computational requirements. Their main weakness is that two geographically distant parties both need to have a key that matches exactly. In the past, keys could be much simpler and still be secure, but with today's computational power, simple keys can be brute-forced very quickly. This means that larger and more complex keys must be used and exchanged. This key exchange is difficult because the key cannot be simple, such as a word, but must be shared in a secure manner. It might be easy to exchange a 4-bit key such as b in hex, but exchanging the 128-bit key 4b36402c5727472d5571373d22675b4b is far more difficult to do securely. This exchange of keys is greatly facilitated by our next subject, asymmetric, or public key, cryptography.

Asymmetric Encryption
Asymmetric cryptography is in many ways completely different than symmetric cryptography. While both are used to keep data from being seen by unauthorized users, asymmetric cryptography uses two keys instead of one. It was invented by Whitfield Diffie and Martin Hellman in 1975. Asymmetric cryptography is more commonly known as public key cryptography. The system uses a pair of keys: a private key that is kept secret and a public key that can be sent to anyone. The system's security relies upon resistance to deducing one key, given the other, and thus retrieving the plaintext from the ciphertext.



RSA
RSA is one of the first public key cryptosystems ever invented. It can be used for both encryption and digital signatures. RSA is named after its inventors, Ron Rivest, Adi Shamir, and Leonard Adleman, and was first published in 1977. This algorithm uses the product of two very large prime numbers and works on the principle of difficulty in

factoring such large numbers. It's best to choose large prime numbers from 100 to 200 digits in length and that are equal in length. These two primes will be P and Q. Randomly choose an encryption key, E, so that E is greater than 1, E is less than P * Q, and E must be odd. E must also be relatively prime to (P – 1) and (Q – 1). Then compute the decryption key D:

D = E–1 mod ((P – 1)(Q – 1))

Now that the encryption key and decryption key have been generated, the two prime numbers can be discarded, but they should not be revealed. To encrypt a message, it should be divided into blocks less than the product of P and Q. Then,

Ci = Mi
E mod (P * Q)

C is the output block of ciphertext matching the block length of the input message, M. To decrypt a message take ciphertext, C, and use this function:

Mi = Ci
D mod (P * Q)

The use of the second key retrieves the plaintext of the message.
This is a simple function, but its security has withstood the test of more than 20 years of analysis. Considering the effectiveness of RSA's security and the ability to have two keys, why are symmetric encryption algorithms needed at all? The answer is speed. RSA in software can be 100 times slower than DES, and in hardware it can be even slower.

Diffie-Hellman
Diffie-Hellman was created in 1976 by Whitfield Diffie and Martin Hellman. This protocol is one of the most common encryption protocols in use today. It plays a role in the electronic key exchange method of the Secure Sockets Layer (SSL) protocol. It is also used by the SSH and IPsec protocols. Diffie-Hellman is important because it enables the sharing of a secret key between two people who have not contacted each other before. The protocol, like RSA, uses large prime numbers to work. Two users agree to two numbers, P and G, with P being a sufficiently large prime number and G being the generator. Both users pick a secret number, a and b. Then both users compute their public number:

User 1 X = Ga mod P, with X being the public number
User 2 Y = Gb mod P, with Y being the public number
The users then exchange public numbers. User 1 knows P, G, a, X, and Y.
User 1 Computes Ka = Ya mod P
User 2 Computes Kb = Xb mod P
With Ka = Kb = K, now both users know the new shared secret K. This is the basic algorithm, and although there have been methods created to strengthen it, Diffie-Hellman

is still in wide use. It remains very effective because of the nature of what it is protecting—a temporary, automatically generated secret key that is good only for a single communication session.

ElGamal
ElGamal can be used for both encryption and digital signatures. Taher ElGamal designed the system in the early 1980s. This system was never patented and is free for use. It is used as the U.S. government standard for digital signatures.

The system is based upon the difficulty of calculating discrete logarithms in a finite field. Three numbers are needed to generate a key pair. User 1 chooses a prime, P, and two random numbers, F and D. F and D should both be less than P.

ECC
Elliptic curve cryptography (ECC) works on the basis of elliptic curves. An elliptic curve is a simple function that is drawn as a gently looping curve on the X,Y plane. They are defined by this equation:

y2 = x3 + ax2 + b

Elliptic curves work because they have a special property—you can add two points on the curve together and get a third point on the curve. For cryptography, the elliptic curve works as a public key algorithm. Users agree on an elliptic curve and a fixed curve point. This information is not a shared secret, and these points can be made public without compromising the security of the system. User 1 then chooses a secret random number, K1, and computes a public key based upon a point on the curve:

P1 = K1 * F

User 2 performs the same function and generates P2. Now user 1 can send user 2 a message
by generating a shared secret:

S = K1 * P2

User 2 can generate the same shared secret independently:

S = K2 * P1

This is true because

K1 * P2 = K1 * (K2 * F) = (K1 * K2) * F = K2 * (K1 * F) = K2 * P1

The security of elliptic curve systems has been questioned, mostly because of lack of analysis. However, all public key systems rely on the difficulty of certain math problems.

It would take a breakthrough in math for any of the mentioned systems to be weakened dramatically, but research has been done about the problems and has shown that the elliptic curve problem has been more resistant to incremental advances. Again, as with all cryptography algorithms, only time will tell how secure they really are.

Asymmetric Encryption Summary
Asymmetric encryption creates the possibility of digital signatures and also corrects the main weakness of symmetric cryptography. The ability to send messages securely without senders and receivers having had prior contact has become one of the basic concerns with secure communication. Digital signatures will enable faster and more efficient exchange of all kinds of documents, including legal documents. With strong algorithms and good key lengths, security can be assured.

Steganography
Steganography, an offshoot of cryptography technology, gets its meaning from the Greek steganos meaning covered. Invisible ink placed on a document hidden by innocuous text is an example of a steganographic message. Another example is a tattoo placed on the top of a person's head, visible only when the person's hair is shaved off. Hidden writing in the computer age relies on a program to hide data inside other data. The most common application is the concealing of a text message in a picture file. The Internet contains multiple billions of image files, allowing a hidden message to be located almost anywhere without being discovered. The nature of the image files also make a hidden message difficult to detect. While it is most common to hide messages inside images, they can also be hidden in video and audio files.

Steganographic encoding can be used in many ways and through many different media. Covering them all is beyond the scope for this short study guide, but we will discuss one of the most common ways to encode into an image file, LSB encoding. LSB, Least Significant Bit, is a method of encoding information into an image while altering the actual visual image as little as possible. A computer image is made up of thousands or millions of pixels, all defined by 1s and 0s. If an image is composed of Red Green Blue (RGB) values, each pixel has an RGB value represented numerically from 0 to 255. For example, 0,0,0 is black, and 255,255,255 is white, which can also be represented as 00000000, 00000000, 00000000 for black and 11111111, 11111111, 11111111 for white. Given a white pixel, editing the least significant bit of the pixel to 11111110, 11111110, 11111110 changes the color. The change in color is undetectable to the human eye, but in a image with a million pixels, this creates a 125KB area in which to store a message.

Cryptography Algorithm Use

The use of cryptographic algorithms grows every day. More and more information becomes digitally encoded and placed online, and all of this data needs to be secured. The best way to do that with current technology is to use encryption. Security is typically defined as a product of five components:

confidentiality, integrity, availability, authentication, and nonrepudiation. Encryption addresses four of these five components: confidentiality, integrity, nonrepudiation, and authentication.

Confidentiality
Confidentiality typically comes to mind when the term security is brought up. Confidentiality is the ability to keep some piece of data a secret. In the digital world, encryption excels at providing confidentiality. Confidentiality is used on stored data and on transmitted data. In both cases, symmetric encryption is favored because of its speed and because some asymmetric algorithms can significantly increase the size of the object being encrypted. In the case of a stored item, a public key is typically unnecessary, as the item is being encrypted to protect it from access by others. In the case of transmitted data, public key cryptography is typically used to exchange the secret key, and then symmetric cryptography is used to ensure the confidentiality of the data being sent.

Asymmetric cryptography does protect confidentiality, but its size and speed make it more efficient at protecting the confidentiality of small units for tasks such as electronic key exchange. In all cases, the strength of the algorithms and the length of the keys ensure the secrecy of the data in question.

Integrity
Integrity is better known as message integrity, and it is a crucial component of message security. When a message is sent, both the sender and recipient need to know that the message was not altered in transmission. This is especially important for legal contracts— recipients need to know that the contracts have not been altered. Signers also need a way to validate that a contract they sign will not be altered in the future. Integrity is provided with one-way hash functions and digital signatures. The hash functions compute the message digests, and this guarantees the integrity of the message by allowing easy testing to determine whether any part of the message has been changed. The message now has a computed function (the hash value) to tell the users to resend the message if it was intercepted and interfered with.

Nonrepudiation
An item of some confusion, the concept of nonrepudiation is actually fairly simple. Nonrepudiation means that the message sender cannot later deny that she sent the message. This is important in electronic exchanges of data, because of the lack of face-to-face meetings. Nonrepudiation is based upon public key cryptography and the principle of only you knowing your private key. The presence of a message signed by you, using your private key, which nobody else should know, is an example of nonrepudiation.

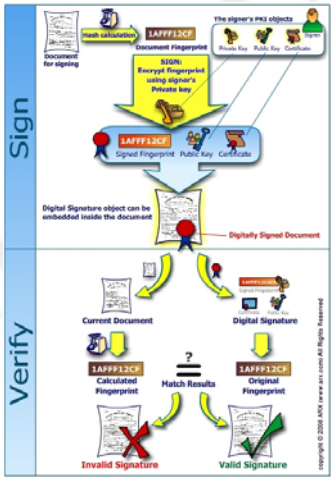
When a third party can check your signature using your public key, that disproves any claim that you were not the one who actually sent the message. Nonrepudiation is tied to asymmetric cryptography and cannot be implemented with symmetric algorithms.

Authentication

Authentication lets you prove you are who you say you are. Authentication is similar to nonrepudiation, except that authentication often occurs as communication begins, not after. Authentication is also typically used in both directions as part of a protocol. Authentication can be accomplished in a multitude of ways, the most basic being the use of a simple password. Every time you sign in to check your e-mail, you authenticate yourself to the server. This process can grow to need two or three identifying factors, such as a password, a token (such as a digital certificate), and a biometric (such as a fingerprint).

Digital Signatures

Digital signatures have been touted as the key to truly paperless document flow, and they do have promise for improving the system. Digital signatures are based on both hashing functions and asymmetric cryptography. Both encryption methods play an important role in signing digital documents.



Unprotected digital documents are very easy for anyone to change. If a document is edited after an individual signs it, it is important that any modification can be detected. To protect against document editing, hashing functions are used to create a digest of the message that is unique and easily reproducible by both parties. This ensures that the message integrity is complete.

Key Escrow

The impressive growth of the use of encryption technology has led to new methods for handling keys. Encryption is adept at hiding secrets, and with computer technology being affordable to everyone, criminals and other ill-willed people began using it to conceal communications and business dealings from law enforcement agencies. Because they could not break the encryption, government agencies began asking for key escrow. Key escrow is a system by which your private key is kept both by you and by the government. This allows people with a court order to retrieve your private key to gain access to anything encrypted with your public key. The data is essentially encrypted by your key and the government key, giving the government access to your plaintext data.

Cryptographic Applications

A few applications can be used to encrypt data conveniently on your personal computer. (This is by no means a complete list of every application.)

Pretty Good Privacy (PGP) is mentioned in this guide because it is a useful protocol suite. Created by Philip Zimmermann in 1991, it passed through several versions that were available for free under a noncommercial license. PGP applications can be plugged into popular e-mail programs to handle the majority of day-to-day encryption tasks using a combination of symmetric and asymmetric encryption protocols. One of the unique

features of PGP is its ability to use both symmetric and asymmetric encryption methods, accessing the strengths of each method and avoiding the weaknesses of each as well. Symmetric keys are used for bulk encryption, taking advantage of the speed and efficiency of symmetric encryption. The symmetric keys are passed using asymmetric methods, capitalizing on the flexibility of this method. PGP is now sold as a commercial application with home and corporate versions. Depending on the version, PGP can perform file encryption, whole disk encryption, and public key encryption to protect e-mail.

TrueCrypt is an open source solution for encryption. It is designed for symmetric disk-based encryption of your files. It features AES ciphers and the ability to create a deniable volume, encryption stored within encryption so that volume cannot be reliably detected. TrueCrypt can perform file encryption and whole disk encryption. Whole disk encryption encrypts the entire hard drive of a computer, including the operating system. FreeOTFE is similar to TrueCrypt. It offers "on-the-fly" disk encryption as an open source freely downloadable application. It can encrypt files up to entire disks with several popular ciphers including AES.

GnuPG or Gnu Privacy Guard is an open source implementation of the OpenPGP standard. This command line–based tool is a public key encryption program designed to protect electronic communications such as e-mail. It operates similar to PGP and includes a method for managing public/private keys.

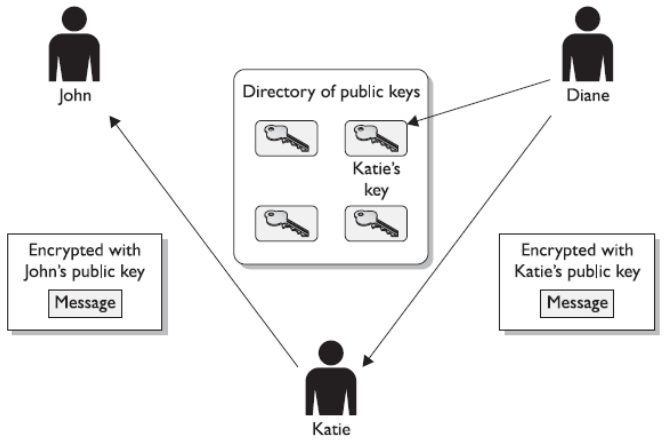
Public Key Infrastructures

Public key infrastructures (PKIs) are becoming a central security foundation for managing identity credentials in many companies. The technology manages the issue of binding public keys and identities across multiple applications. The other approach, without PKIs, is to implement many different security solutions and hope for interoperability and equal levels of protection.

PKIs comprise components that include certificates, registration and certificate authorities, and a standard process for verification. PKI is about managing the sharing of trust and using a third party to vouch for the trustworthiness of a claim of ownership over a credential document, called a certificate.

The Basics of Public Key Infrastructures

A PKI provides all the components necessary for different types of users and entities to be able to communicate securely and in a predictable manner. A PKI is made up of hardware, applications, policies, services, programming interfaces, cryptographic algorithms, protocols, users, and utilities. These components work together to allow communication to take place using public key cryptography and asymmetric keys for digital signatures, data encryption, and integrity. Although many different applications and protocols can provide the same type of functionality, constructing and implementing a PKI boils down to establishing a level of trust.

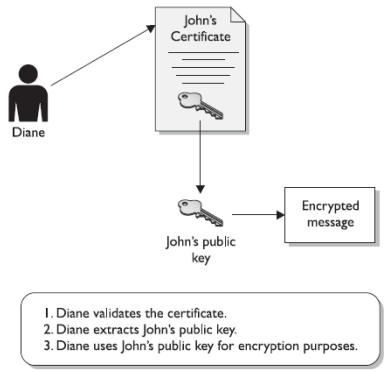


If, for example, John and Diane want to communicate securely, John can generate his own public/private key pair and send his public key to Diane, or he can place his public key in a directory that is available to everyone. If Diane receives John's public key, either from him or from a public directory, how does she know it really came from John? Maybe another individual is masquerading as John and replaced John's public key with her own. If this took place, Diane would believe that her messages could be read only by John and that the replies were actually from him. However, she would actually be communicating with Katie. What is needed is a way to verify an individual's identity, to ensure that a person's public key is bound to their identity and thus ensure that the previous scenario (and others) cannot take place.

In PKI environments, entities called registration authorities and certificate authorities (CAs) provide services similar to those of the Department of Motor Vehicles (DMV). When John goes to register for a driver's license, he has to prove his identity to the DMV by providing his passport, birth certificate, or other identification documentation. If the DMV is satisfied with the proof John provides (and John passes a driving test), the DMV will create a driver's license that can then be used by John to prove his identity. Whenever John needs to identify himself, he can show his driver's license. Although

many people may not trust John to identify himself truthfully, they do trust the third party, the DMV.

What does the "infrastructure" in "public key infrastructure" really mean? An infrastructure provides a sustaining groundwork upon which other things can be built. So an infrastructure works at a low level to provide a predictable and uniform environment that allows other higher level technologies to work together through uniform access points. The environment that the infrastructure provides allows these higher level applications to communicate with each other and gives them the underlying tools to carry out their tasks.



Certificate Authorities

The CA is the trusted authority that certifies individuals' identities and creates electronic documents indicating that individuals are who they say they are. The electronic document is referred to as a digital certificate, and it establishes an association between the subject's identity and a public key. The private key that is paired with the public key in the certificate is stored separately. It is important to safeguard the private key, and it typically never leaves the machine or device where it was created.

The CA is more than just a piece of software, however; it is actually made up of the software, hardware, procedures, policies, and people who are involved in validating individuals' identities and generating the certificates. This means that if one of these components is compromised, it can negatively affect the CA overall and can threaten the integrity of the certificates it produces.

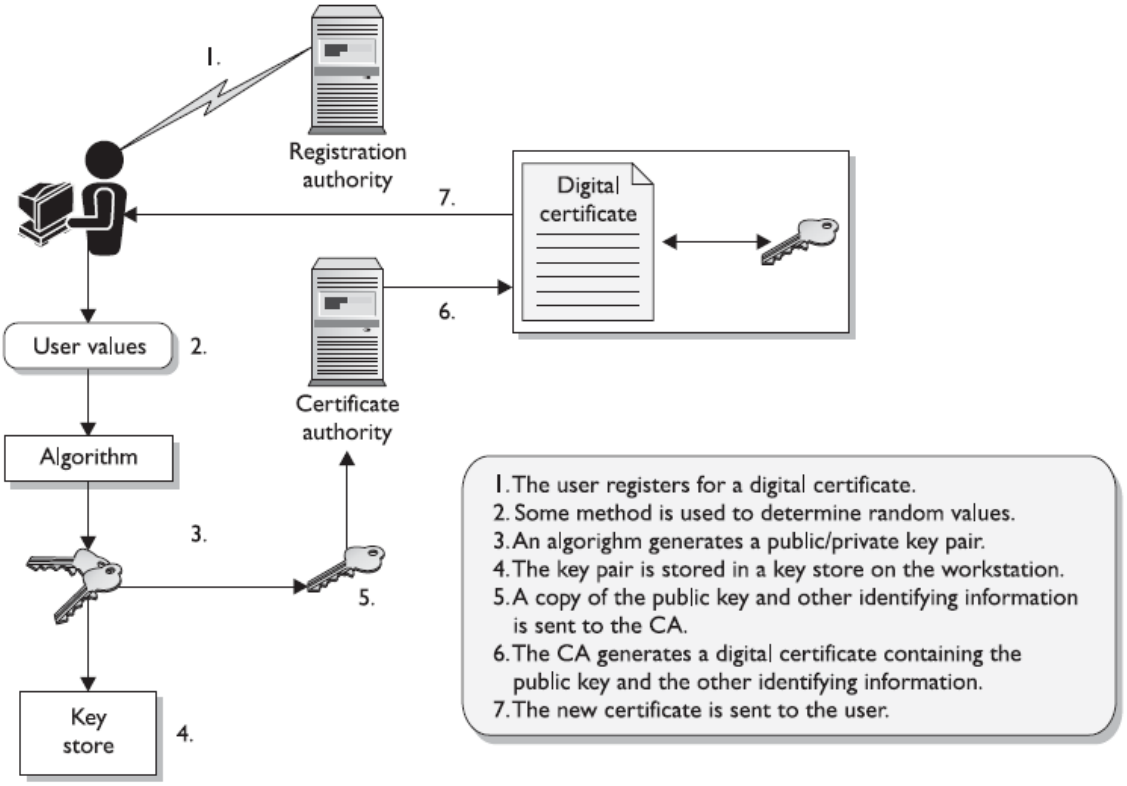
Every CA should have a certification practices statement (CPS) that outlines how identities are verified; the steps the CA follows to generate, maintain, and transmit certificates; and why the CA can be trusted to fulfill its responsibilities. It describes how keys are secured, what data is placed within a digital certificate, and how revocations will be handled. If a company is going to use and depend on a public CA, the company's security officers, administrators, and legal department should review the CA's entire CPS to ensure that it will properly meet the company's needs, and to make sure that the level of security claimed by the CA is high enough for their use and environment. A critical aspect of a PKI is the trust between the users and the CA, so the CPS should be reviewed and understood to ensure that this level of trust is warranted.

Registration Authorities

The registration authority (RA) is the component that accepts a request for a digital certificate and performs the necessary steps of registering and authenticating the person requesting the certificate. The authentication requirements differ depending on the type of certificate being requested.

The types of certificates available can vary between different CAs, but usually at least three different types are available, and they are referred to as classes:

Class 1 A Class 1 certificate is usually used to verify an individual's identity through e-mail. A person who receives a Class 1 certificate can use his public/ private key pair to digitally sign e-mail and encrypt message contents.
Class 2 A Class 2 certificate can be used for software signing. A software vendor would register for this type of certificate so it could digitally sign its software. This provides integrity for the software after it is developed and released, and it allows the receiver of the software to verify from where the software actually came.
Class 3 A Class 3 certificate can be used by a company to set up its own CA, which will allow it to carry out its own identification verification and generate certificates internally.



Each higher class of certificate can carry out more powerful and critical tasks than the one before it. This is why the different classes have different requirements for proof of identity. If you want to receive a Class 1 certificate, you may only be asked to provide your name, e-mail address, and physical address. For a Class 2 certification, you may need to provide the RA with more data, such as your driver's license, passport, and company information that can be verified. To obtain a Class 3 certificate, you will be asked to provide even more information and most likely will need to go to the RA's office for a face-to-face meeting. Each CA will outline the certification classes it provides

and the identification requirements that must be met to acquire each type of certificate In most situations, when a user requests a Class 1 certificate, the registration process will require the user to enter specific information into a web-based form. The web page will have a section that accepts the user's public key, or it will step the user through creating a public/private key pair, which will allow the user to choose the size of the keys to be created. Once these steps have been completed, the public key is attached to the certificate registration form and both are forwarded to the RA for processing. The RA is responsible only for the registration process and cannot actually generate a certificate.

Once the RA is finished processing the request and verifying the individual's identity, the RA will send the request to the CA. The CA will use the RA-provided information to generate a digital certificate, integrate the necessary data into the certificate fields (user identification information, public key, validity dates, proper use for the key and certificate, and so on), and send a copy of the certificate to the user.

Local Registration Authorities

A local registration authority (LRA) performs the same functions as an RA, but the LRA is closer to the end users. This component is usually implemented in companies that have their own internal PKIs and have distributed sites. Each site has users that need RA services, so instead of requiring them to communicate with one central RA, each site can have its own LRA. This reduces the amount of traffic that would be created by several users making requests across wide area network (WAN) lines. The LRA will perform identification, verification, and registration functions. It will then send the request, along with the user's public key, to a centralized CA so that the certificate can be generated.

It acts as an interface between the users and the CA. LRAs simplify the RA/CA process for entities that desire certificates only for in-house use.

Certificate Repositories

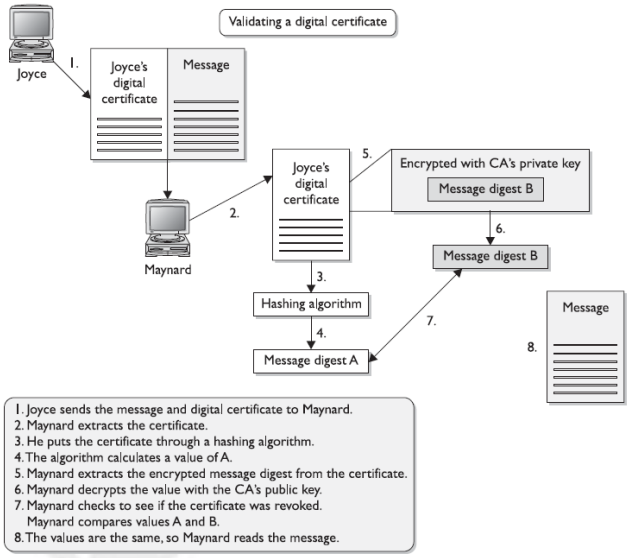
Once the requestor's identity has been proven, a certificate is registered with the public side of the key pair provided by the requestor. Public keys must be available to anybody who requires them to communicate within a PKI environment. These keys, and their corresponding certificates, are usually held in a publicly available repository. Repository is a general term that describes a centralized directory that can be accessed by a subset of individuals. The directories are usually Lightweight Directory Access Protocol (LDAP)–compliant, meaning that they can be accessed and searched via the LDAP.

When an individual initializes communication with another, the sender can send her certificate and public key to the receiver, which will allow the receiver to communicate with the sender using encryption or digital signatures (or both) without needing to track down the necessary public key in a certificate repository. This is equivalent to the sender saying, "If you would like to encrypt any future messages you send to me, or if you would like the ability to verify my digital signature, here are the necessary components."

But if a person wants to encrypt the first message sent to the receiver, the sender will need to find the receiver's public key in a certificate repository.

Trust and Certificate Verification

We need to use a PKI if we do not automatically trust individuals we do not know. Security is about being suspicious and being safe, so we need a third party that we do trust to vouch for the other individual before confidence can be instilled and sensitive communication can take place. But what does it mean that we trust a CA, and how can we use this to our advantage?



When a user chooses to trust a CA, she will download that CA's digital certificate and public key, which will be stored on her local computer. Most browsers have a list of CAs configured to be trusted by default, so when a user installs a new web browser, several of the most well-known and most trusted CAs will be trusted without any change of settings.

In the Microsoft CAPI environment, the user can add and remove CAs from this list as needed. In production environments that require a higher degree of protection, this list will be pruned, and possibly the only CAs listed will be the company's internal CAs. This ensures that digitally signed software will be automatically installed only if it was signed by the company's CA. Other products, such as Entrust, use centrally controlled policies to determine which CAs are to be trusted instead of expecting the user to make these critical decisions.

Digital Certificates

A digital certificate binds an individual's identity to a public key, and it contains all the information a receiver needs to be assured of the identity of the public key owner. After an RA verifies an individual's identity, the CA generates the digital certificate, but how does the CA know what type of data to insert into the certificate? The certificates are created and formatted based on the X.509 standard, which outlines the necessary fields of a certificate and the possible values that can be inserted into the fields. As of this writing, X.509 version 3 is the most current version of the standard. X.509 is a standard of the International Telecommunication Union (www.itu.int). The IETF's Public-Key Infrastructure (X.509), or PKIX, working group has adapted the X.509 standard to the more flexible organization of the Internet, as specified in RFC 3280, and is commonly referred to as PKIX for Public Key Infrastructure (X.509).

The following fields are included within a X.509 digital certificate:

- **Version number** Identifies the version of the X.509 standard that was followed to create the certificate; indicates the format and fields that can be used.
- **Subject** Specifies the owner of the certificate.
- **Public key** Identifies the public key being bound to the certified subject; also identifies the algorithm used to create the private/public key pair.
- **Issuer** Identifies the CA that generated and digitally signed the certificate.
- **Serial number** Provides a unique number identifying this one specific certificate issued by a particular CA.
- **Validity** Specifies the dates through which the certificate is valid for use.
- **Certificate usage** Specifies the approved use of certificate, which dictates intended use of this public key.
- **Signature algorithm** Specifies the hashing and digital signature algorithms used to digitally sign the certificate.
- **Extensions** Allow additional data to be encoded into the certificate to expand the functionality of the certificate. Companies can customize the use of certificates within their environments by using these extensions. X.509 version 3 has extended the extension possibilities.

The subject of a certificate is commonly a person, but it does not have to be. The subject can be a network device (router, web server, firewall, and so on), an application, a department, a company, or a person. Each has its own identity that needs to be verified and proven to another entity before secure, trusted communication can be initiated.

If a network device is using a certificate for authentication, the certificate may contain the network address of that device. This means that if the certificate has a network address of 10.0.0.1, the receiver will compare this to the address from which it received the certificate to make sure a man-in-the-middle attack is not being attempted.

Certificate Attributes

Four main types of certificates are used:
- End-entity certificates
- CA certificates
- Cross-certification certificates
- Policy certificates

End-entity certificates are issued by a CA to a specific subject, such as Joyce, the Accounting department, or a firewall. An end-entity certificate is the identity document provided by PKI implementations.

A CA certificate can be self-signed, in the case of a standalone or root CA, or it can be issued by a superior CA within a hierarchical model. The superior CA gives the authority

and allows the subordinate CA to accept certificate requests and generate the individual certificates itself. This may be necessary when a company needs to have multiple internal CAs, and different departments within an organization need to have their own CAs servicing their specific end-entities in their sections. In these situations, a representative from each department requiring a CA registers with the higher trusted CA and requests a Certificate Authority certificate. Cross-certificates, or cross-certification certificates, are used when independent CAs establishes peer-to-peer trust relationships. Simply put, they are a mechanism through which one CA can issue a certificate allowing its users to trust another CA. Within sophisticated CAs used for high-security applications, a mechanism is required to provide centrally controlled policy information to PKI clients. This is often done by placing the policy information in a policy certificate.

Certificate Extensions
Certificate extensions allow for further information to be inserted within the certificate, which can be used to provide more functionality in a PKI implementation. Certificate extensions can be standard or private. Standard certificate extensions are implemented for every PKI implementation. Private certificate extensions are defined for specific organizations (or domains within one organization), and they allow companies to further define different, specific uses for digital certificates to best fit their business needs. Several different extensions can be implemented, one being key usage extensions, which dictate how the public key that is held within the certificate can be used. Remember that public keys can be used for different functions: symmetric key encryption, data encryption, verifying digital signatures, and more. Following are some key examples of certificate extension:

- **DigitalSignature** The key used to verify a digital signature
- **KeyEncipherment** The key used to encrypt other keys used for secure key distribution
- **DataEncipherment** The key used to encrypt data, which cannot be used to encrypt other keys
- **CRLSign** The key used to verify a CA signature on a revocation list
- **KeyCertSign** The key used to verify CA signatures on certificates
- **NonRepudiation** The key used when a nonrepudiation service is being provided

Critical and Noncritical Extensions
Certificate extensions are considered either critical or noncritical, which is indicated by a specific flag within the certificate itself. When this flag is set to critical, it means that the extension must be understood and processed by the receiver. If the receiver is not configured to understand a particular extension marked as critical, and thus cannot process it properly, the certificate cannot be used for its proposed purpose. If the flag does not indicate that the extension is critical, the certificate can be used for the intended purpose, even if the receiver does not process the appended extension.

So how does this work? When an extension is marked as critical, it means that the CA is certifying the key for only that specific purpose. If Joe receives a certificate with a

DigitalSignature key usage extension and the critical flag is set, Joe can use the public key only within that certificate to validate digital signatures, and no more. If the extension was marked as noncritical, the key can be used for purposes outside of those listed in the extensions, so in this case it is up to Joe (and his applications) to decide how the key will be used.

Certificate Lifecycles
Keys and certificates should have lifetime settings that will force the user to register for a new certificate after a certain amount of time. Determining the proper length of these lifetimes is a trade-off: Shorter lifetimes limit the ability of attackers to crack them, but longer lifetimes lower system overhead. More sophisticated PKI implementations perform automated and often transparent key updates to avoid the time and expense of having users register for new certificates when old ones expire. This means that the certificate and key pair has a lifecycle that must be managed. Certificate management involves administrating and managing each of these phases, including registration, certificate and key generation, renewal, and revocation.

Registration and Generation
A key pair (public and private keys) can be generated locally by an application and stored in a local key store on the user's workstation. The key pair can also be created by a central key-generation server, which will require secure transmission of the keys to the user. The key pair that is created on the centralized server can be stored on the user's workstation or on the user's smart card, which will allow for more flexibility and mobility. In most modern PKI implementations, users have two key pairs. One key pair is often generated by a central server and used for encryption and key transfers. This allows the corporate PKI to retain a copy of the encryption key pair for recovery, if necessary. The second key pair, a digital signature key pair, is usually generated by the user to make sure that she is the only one with a copy of the private key. Nonrepudiation can be challenged if there is any doubt about someone else obtaining a copy of an individual's signature private key. If the key pair was created on a centralized server, that could weaken the case that the individual was the only one who had a copy of her private key. If a copy of a user's signature private key is stored anywhere other than in her possession, or if there is a possibility of someone obtaining the user's key, then true nonrepudiation cannot be provided.

Renewal
The certificate itself has its own lifetime, which can be different than the key pair's lifetime. The certificate's lifetime is specified by the validity dates inserted into the digital certificate. These are beginning and ending dates indicating the time period during which the certificate is valid. The certificate cannot be used before the start date, and once the end date is met, the certificate is expired and a new certificate will need to be issued.

A renewal process is different from the registration phase in that the RA assumes that the individual has already successfully completed one registration round. If the certificate has

not actually been revoked, the original keys and certificate can be used to provide the necessary authentication information and proof of identity for the renewal phase.

Revocation
A certificate can be revoked when its validity needs to be ended before its actual expiration date is met, and this can occur for many reasons: for example, a user may have lost a laptop or a smart card that stored a private key, an improper software implementation may have been uncovered that directly affected the security of a private key, a user may have fallen victim to a social engineering attack and inadvertently given up a private key, data held within the certificate may no longer apply to the specified individual, or perhaps an employee left a company and should not be identified as a member of an in-house PKI any longer. In the last instance, the certificate, which was bound to the user's key pair, identified the user as an employee of the company, and the administrator would want to ensure that the key pair could not be used in the future to validate this person's affiliation with the company. Revoking the certificate does this.

If any of these things happen, a user's private key has been compromised or should no longer be mapped to the owner's identity. A different individual may have access to that user's private key and could use it to impersonate and authenticate as the original user. If the impersonator used the key to digitally sign a message, the receiver would verify the authenticity of the sender by verifying the signature by using the original user's public key, and the verification would go through perfectly—the receiver would believe it came from the proper sender and not the impersonator. If receivers could look at a list of certificates that had been revoked before verifying the digital signature, however, they would know not to trust the digital signatures on the list. Because of issues associated with the private key being compromised, revocation is permanent and final—once revoked, a certificate cannot be reinstated. If these were allowed and a user revoked his certificate, the unauthorized holder of the private key could use it to restore the certificate validity.

CRL Distribution
CRL files can be requested by individuals who need to verify and validate a newly received certificate, or the files can be periodically pushed down (sent) to all users participating within a specific PKI. This means the CRL can be pulled (downloaded) by individual users when needed or pushed down to all users within the PKI on a timed interval.

The actual CRL file can grow substantially, and transmitting this file and requiring PKI client software on each workstation to save and maintain it can use a lot of resources, so the smaller the CRL is, the better. It is also possible to first push down the full CRL, and after that initial load, the following CRLs pushed down to the users are delta CRLs, meaning that they contain only the changes to the original or base CRL. This can greatly reduce the amount of bandwidth consumed when updating CRLs

Suspension

Instead of being revoked, a certificate can be suspended, meaning it is temporarily put on hold. If, for example, Bob is taking an extended vacation and wants to ensure that his certificate will not be used during that time, he can make a suspension request to the CA. The CRL would list this certificate and its serial number, and in the field that describes why the certificate is revoked, it would instead indicate a hold state. Once Bob returns to work, he can make a request to the CA to remove his certificate from the list.

Key Destruction
Key pairs and certificates have set lifetimes, meaning that they will expire at some specified time. It is important that the certificates and keys are properly destroyed when that time comes, wherever the keys are stored (on users' workstations, centralized key servers, USB token devices, smart cards, and so on).

Centralized or Decentralized Infrastructures
Keys used for authentication and encryption within a PKI environment can be generated in a centralized or decentralized manner. In a decentralized approach, software on individual computers generates and stores cryptographic keys local to the systems themselves. In a centralized infrastructure, the keys are generated and stored on a central server, and the keys are transmitted to the individual systems as needed. You might choose one type over the other for several reasons.

If a company uses an asymmetric algorithm that is resource-intensive to generate the public/private key pair, and if large (and resource-intensive) key sizes are needed, then the individual computers may not have the necessary processing power to produce the keys in an acceptable fashion. In this situation, the company can choose a centralized approach in which a very high-end server with powerful processing abilities is used, probably along with a hardware-based random number generator.

Hardware Storage Devices
PKIs can be constructed in software without special cryptographic hardware, and this is perfectly suitable for many environments. But software can be vulnerable to viruses, hackers, and hacking. If a company requires a higher level of protection than a purely software-based solution can provide, several hardware-based solutions are available.

Private Key Protection
Although a PKI implementation can be complex, with many different components and options, a critical concept common to all PKIs must be understood and enforced: the private key needs to stay private. A digital signature is created solely for the purpose of proving who sent a particular message by using a private key. This rests on the assumption that only one person has access to this private key. If an imposter obtains a user's private key, authenticity and nonrepudiation can no longer be claimed or proven.

When a private key is generated for the first time, it must be stored somewhere for future use. This storage area is referred to as a key store, and it is usually created by the application registering for a certificate, such as a web browser, smart card software, or

other application. In most implementations, the application will prompt the user for a password, which will be used to create an encryption key that protects the key store. So, for example, if Cheryl used her web browser to register for a certificate, her private key would be generated and stored in the key store. Cheryl would then be prompted for a password, which the software would use to create a key that will encrypt the key store.

When Cheryl needs to access this private key later that day, she will be prompted for the same password, which will decrypt the key store and allow her access to her private key.

Key Recovery
One individual could have one, two, or many key pairs that are tied to his or her identity. That is because users can have different needs and requirements for public/private key pairs. As mentioned earlier, certificates can have specific attributes and usage requirements dictating how their corresponding keys can and cannot be used. For example, David can have one key pair he uses to encrypt and transmit symmetric keys. He can also have one key pair that allows him to encrypt data and another key pair to perform digital signatures. David can also have a digital signature key pair for his work related activities and another pair for personal activities, such as e-mailing his friends.

These key pairs need to be used only for their intended purposes, and this is enforced through certificate attributes and usage values.

Key Escrow
Key recovery and key escrow are terms that are often used interchangeably, but they actually describe two different things. You should not use them interchangeably after you have read this section. Key recovery is a process that allows for lost keys to be recovered. Key escrow is a process of giving keys to a third party so that they can decrypt and read sensitive information when this need arises. Key escrow almost always pertains to handing over encryption keys to the government, or to another higher authority, so that the keys can be used to collect evidence during investigations. A key pair used in a person's place of work may be required to be escrowed by the employer for obvious reasons. First, the keys are property of the enterprise, issued to the worker for use. Second, the firm may have need for them after an employee leaves the firm.

Public Certificate Authorities
An individual or company may decide to rely on a CA that is already established and being used by many other individuals and companies—this would be a public CA. A company, on the other hand, may decide that it needs its own CA for internal use, which gives the company more control over the certificate registration and generation process and allows it to configure items specifically for its own needs. This second type of CA is referred to as a private CA (or in-house CA).

A public CA specializes in verifying individual identities and creating and maintaining their certificates. These companies issue certificates that are not bound to specific companies or intercompany departments. Instead, their services are to be used by a larger

and more diversified group of people and organizations. If a company uses a public CA, the company will pay the CA organization for individual certificates and for the service of maintaining these certificates. Some examples of public CAs are VeriSign (including GeoTrust and thawte), Entrust, and Go Daddy.

One advantage of using a public CA is that it is usually well known and easily accessible to many people. Most web browsers have a list of public CAs installed and configured by default, along with their corresponding root certificates. This means that if you install a web browser on your computer, it is already configured to trust certain CAs, even though you might have never heard of them before. So, if you receive a certificate from Bob, and his certificate was digitally signed by a CA listed in your browser, you can automatically trust the CA and can easily walk through the process of verifying Bob's certificate. This has raised some eyebrows among security professionals, however, since trust is installed by default, but the industry has deemed this is a necessary approach that provides users with transparency and increased functionality. Users can remove these CAs from their browser list if they want to have more control over who their system trusts and who it doesn't.

In-house Certificate Authorities
An in-house CA is implemented, maintained, and controlled by the company that implemented it. This type of CA can be used to create certificates for internal employees, devices, applications, partners, and customers. This approach gives the company complete control over how individuals are identified, what certification classifications are created, who can and cannot have access to the CA, and how the certifications can be used.

In-house CAs also provides more flexibility for companies, which often integrate them into current infrastructures and into applications for authentication, encryption, and nonrepudiation purposes. If the CA is going to be used over an extended period of time, this can be a cheaper method of generating and using certificates than having to purchase them through a public CA.

Outsourced Certificate Authorities
The last available option for using PKI components within a company is to outsource different parts of it to a specific service provider. Usually, the more complex parts are outsourced, such as the CA, RA, CRL, and key recovery mechanisms. This occurs if a company does not have the necessary skills to implement and carry out a full PKI environment. An outsourced CA is different from a public CA in that it provides dedicated services, and possibly equipment, to an individual company. A public CA, in contrast, can be used by hundreds or thousands of companies—the CA doesn't maintain specific servers and infrastructures for individual companies.

Although outsourced services might be easier for your company to implement, you need to review several factors before making this type of commitment. You need to determine what level of trust the company is willing to give to the service provider and what level

of risk it is willing to accept. Often a PKI and its components serve as large security components within a company's enterprise and allowing a third party to maintain the PKI can introduce too many risks and liabilities that your company is not willing to undertake. The liabilities the service provider is willing to accept, security precautions and procedures the outsourced CAs provide, and the surrounding legal issues need to be examined before this type of agreement is made.

Physical Security

Physical security is an important topic for businesses dealing with the security of information systems. Businesses are responsible for securing their profitability, which requires a combination of several aspects: They need to secure employees, product inventory, trade secrets, and strategy information. These and other important assets affect the profitability of a company and its future survival. Companies therefore perform many activities to attempt to provide physical security—locking doors, installing alarm systems, using safes, posting security guards, setting access controls, and more.

Most companies today have committed a large amount of effort into network security and information systems security. In this chapter, you will learn about how these two security efforts are linked, and you'll learn several methods by which companies can minimize their exposure to physical security events that can diminish their network security.

The Security Problem

The problem that faces professionals charged with securing a company's network can be stated rather simply: Physical access negates all other security measures. No matter how impenetrable the firewall and intrusion detection system (IDS), if an attacker can find a way to walk up to and touch a server, he can break into it. The more remarkable thing is that gaining physical access to a number of machines is not that difficult.

Consider that most network security measures are, from necessity, directed at protecting a company from the Internet. This fact results in a lot of companies allowing any kind of traffic on the local area network (LAN). So if an attacker attempts to gain access to a server over the Internet and fails, he may be able to gain physical access to the receptionist's machine, and by quickly compromising it, he can use it as a remotely controlled zombie to attack what he is really after. Physically securing information assets doesn't mean just the servers; it means protecting the physical access to all the organization's computers and its entire network infrastructure.

Physical access to a corporation's systems can allow an attacker to perform a number of interesting activities, starting with simply plugging into an open Ethernet jack. The advent of handheld devices with the ability to run operating systems with full networking support has made this attack scenario even more feasible. Prior to handheld devices, the attacker would have to work in a secluded area with dedicated access to the Ethernet for a time. The attacker would sit down with a laptop and run a variety of tools against the network, and working internally typically put the attacker behind the firewall and IDS. Today's capable PDAs can assist these efforts by allowing attackers to place the small device onto the network to act as a wireless bridge. The attacker can then use a laptop to attack a network remotely via the bridge from outside the building. If power is available near the Ethernet jack, this type of attack can also be accomplished with an off-the-shelf

access point. The attacker's only challenge is finding an Ethernet jack that isn't covered by furniture or some other obstruction.

Drive imaging is the process of copying the entire contents of a hard drive to a single file on a different media. This process is often used by people who perform forensic investigations of computers. Typically, a bootable media is used to start the computer and load the drive imaging software. This software is designed to make a bit-by-bit copy of the hard drive to a file on another media, usually another hard drive or CD-R/ DVD-R media. Drive imaging is used in investigations to make an exact copy that can be observed and taken apart, while keeping the original exactly as it was for evidence purposes.

From an attacker's perspective, drive imaging software is useful because it pulls all information from a computer's hard drive while still leaving the machine in its original state. The information contains every bit of data that was on this computer: any locally stored documents, locally stored e-mails, and every other piece of information that the hard drive contained. This data could be very valuable if the machine held sensitive information about the company.

Physical access is the most common way of imaging a drive, and the biggest benefit for the attacker is that drive imaging leaves absolutely no trace of the crime. While you can do very little to prevent drive imaging, you can minimize its impact. The use of encryption even for a few important files will provide protection. Full encryption of the drive will protect all files stored on it. Alternatively, placing files on a centralized file server will keep them from being imaged from an individual machine, but if an attacker is able to image the file server, the data will be copied.

Physical access can negate almost all the security that the network attempts to provide. Considering this, you must determine the level of physical access that attackers might obtain. Of special consideration are persons with authorized access to the building but who are not authorized users of the systems. Janitorial personnel and others have authorized access to many areas, but they do not have authorized system access. An attacker could pose as one of these individuals or attempt to gain access to the facilities through them.

Physical Security Safeguards

While it is difficult, if not impossible, to be totally secure, many steps can be taken to mitigate the risk to information systems from a physical threat. The following sections discuss policies and procedures as well as access control methods.

Walls and Guards

The primary defense against a majority of physical attacks is the barriers between the assets and a potential attacker—walls and doors. Some organizations also employ full or part-time private security staff to attempt to protect their assets. These barriers provide the foundation upon which all other security initiatives are based, but the security must be

designed carefully, as an attacker has to find only a single gap to gain access. Walls may have been one of the first inventions of man. Once he learned to use natural obstacles such as mountains to separate him from his enemy, he next learned to build his own mountain for the same purpose. Hadrian's Wall in England, the Great Wall of China, and the Berlin Wall are all famous examples of such basic physical defenses.

In the case of information assets, as a general rule the most valuable assets are contained on company servers. To protect the physical servers, you must look in all directions: Doors and windows should be safeguarded and a minimum number of each should be used in a server room. Less obvious entry points should also be considered: Is a drop ceiling used in the server room? Do the interior walls extend to the actual roof, raised floors, or crawlspaces? Access to the server room should be limited to the people who need access, not to all employees of the organization. If you are going to use a wall to protect an asset, make sure no obvious holes appear in that wall.

Security personnel can be helpful in securing information assets, but proper protection must be provided. Security guards are typically not computer security experts, so they need to be educated about network security as well as physical security involving users. They are the company's eyes and ears for suspicious activity, so the network security department needs to train them to notice suspicious network activity as well. Multiple extensions ringing in sequence during the night, computers rebooting all at once, or strange people parked in the parking lot with laptop computers are all indicators of a network attack that might be missed. Many traditional physical security tools such as access controls and CCTV camera systems are transitioning from closed hardwired systems to Ethernet- and IP-based systems. This transition opens up the devices to network attacks traditionally performed on computers. With physical security systems being implemented using the IP network, everyone in physical security must become smarter about network security.

Policies and Procedures

A policy's effectiveness depends on the culture of an organization, so all of the policies mentioned here should be followed up by functional procedures that are designed to implement them. Physical security policies and procedures relate to two distinct areas: those that affect the computers themselves and those that affect users.

To mitigate the risk to computers, physical security needs to be extended to the computers themselves. To combat the threat of boot disks, the simplest answer is to remove or disable floppy drives from all desktop systems that do not require them. The continued advance of hard drive capacity has pushed file sizes beyond what floppies can typically hold. LANs with constant Internet connectivity have made network services the focus of how files are moved and distributed. These two factors have reduced floppy usage to the point where computer manufacturers are making floppy drives accessory options instead of standard features. The second boot device to consider is the CD-ROM/DVD-ROM drive. This device can probably also be removed from or disabled on a number of machines. A DVD can not only be used as a boot device, but it can be

exploited via the autorun feature that some operating systems support. Autorun was designed as a convenience for users, so that when a CD containing an application is inserted, the computer will instantly prompt for input versus having to explore the CD filesystem and find the executable file. Unfortunately, since the autorun file runs an executable, it can be programmed to do anything an attacker wants. If autorun is programmed maliciously, it could run an executable that installs malicious code that could allow an attacker to later gain remote control of the machine.

To prevent an attacker from editing the boot order, BIOS passwords should be set. These passwords should be unique to the machine and, if possible, complex, using multiple uppercase and lowercase characters as well as numerics. Considering how often these passwords will be used, it is a good idea to list them all in an encrypted file so that a master passphrase will provide access to them.

The most interesting of these, for security purposes, are the USB flash memory– based storage devices. USB drive keys, which are basically flash memory with a USB interface in a device about the size of your thumb, provide a way to move files easily from computer to computer. When plugged into a USB port, these devices auto-mount and behave like any other drive attached to the computer. Their small size and relatively large capacity, coupled with instant read-write ability, present security problems.

They can easily be used by an individual with malicious intent to conceal the removal of files or data from the building or to bring malicious files into the building and onto the company network.

In addition, well-intentioned users could accidentally introduce malicious code from USB devices by using them on an infected home machine and then bringing the infected device to the office, allowing the malware to bypass perimeter protections and possibly infect the organization. If USB devices are allowed, aggressive virus scanning should be implemented throughout the organization. The devices can be disallowed via Active Directory settings or with a Windows registry key entry. They could also be disallowed by unloading and disabling the USB drivers from user's machines, which will stop all USB devices from working—however, doing this can create more trouble if users have USB keyboards and mice. Editing the registry key is probably the most effective solution for users who are not authorized to use these devices. Users who do have authorization for USB drives must be educated about the potential dangers of their use.

Users should be briefed on the proper departments or personnel to contact when they suspect a security violation. Users can perform one of the most simple, yet important, information security tasks: locking a workstation immediately before they step away from it. While a locking screensaver is a good policy, setting it to less than 15 minutes is often counter-productive to active use on the job. An attacker only needs to be lucky enough to catch a machine that has been left alone for 5 minutes. It is also important to know about workers typically overlooked in the organization. New hires should undergo a background check before being given access to network resources. This policy should

also apply to all personnel who will have unescorted physical access to the facility, including janitorial and maintenance workers.

Access Controls and Monitoring

Access control means control of doors and entry points. The design and construction of all types of access control systems as well as the physical barriers to which they are most complementary are fully discussed in other texts. Here, we explore a few important points to help you safeguard the information infrastructure, especially where it meets with the physical access control system. This section talks about layered access systems, as well as electronic door control systems. It also discusses closed circuit television (CCTV) systems and the implications of different CCTV system types.

Locks have been discussed as a primary element of security. Although locks have been used for hundreds of years, their design has not changed much: a metal "token" is used to align pins in a mechanical device. As all mechanical devices have tolerances, it is possible to sneak-through these tolerances by "picking" the lock.

Tip
A mantrap door arrangement can prevent unauthorized people from following authorized users through an access controlled door, which is also known as "tailgating."

Layered access is an important concept in security. It is often mentioned in conversations about network security perimeters, but in this guide it relates to the concept of physical security perimeters. To help prevent an attacker from gaining access to important assets, these assets should be placed inside multiple perimeters. Servers should be placed in a separate secure area, ideally with a separate authentication mechanism. For example, if an organization has an electronic door control system using *contactless access cards,* a combination of the card and a separate PIN code would be required to open the door to the server room. Access to the server room should be limited to staff with a legitimate need to work on the servers. To layer the protection, the area surrounding the server room should also be limited to people who need to work in that area.

Many organizations use electronic access control systems to control the opening of doors. Doorways are electronically controlled via electronic door strikes and magnetic locks. These devices rely on an electronic signal from the control panel to release the mechanism that keeps the door closed. These devices are integrated into an access control system that controls and logs entry into all the doors connected to it, typically through the use of *access tokens*. Security is improved by having a centralized system that can instantly grant or refuse access based upon a token that is given to the user. This kind of system also logs user access, providing non-repudiation of a specific user's presence in a controlled environment. The system will allow logging of personnel entry, auditing of personnel movements, and real-time monitoring of the access controls.

One caution about these kinds of systems is that they usually work with a software package that runs on a computer, and as such this computer should not be attached to the company network. While attaching it to the network can allow easy administration, the

last thing you want is for an attacker to have control of the system that allows physical access to your facility. With this control, an attacker could input the ID of a badge that she owns, allowing full legitimate access to an area the system controls. Another problem with such a system is that it logs only the person who initially used the card to open the door—so no logs exist for doors that are propped open to allow others access, or of people "tailgating" through a door opened with a card. The implementation of a *mantrap* is one way to combat this function. A mantrap comprises two doors closely spaced that require the user to card through one and then the other sequentially. Mantraps make it nearly impossible to trail through a doorway undetected—if you happen to catch the first door, you will be trapped in by the second door.

CCTVs are similar to the door control systems—they can be very effective, but how they are implemented is an important consideration. The use of CCTV cameras for surveillance purposes dates back to at least 1961, when the London Transport train station installed cameras. The development of smaller camera components and lower costs has caused a boon in the CCTV industry since then.

Environmental Controls

While the confidentiality of information is important, so is its availability. Sophisticated environmental controls are needed for current data centers. Fire suppression is also an important consideration when dealing with information systems.

Heating ventilating and air conditioning (HVAC) systems are critical for keeping data centers cool, because typical servers put out between 1000 and 2000 BTUs of heat. Enough servers in a confined area will create conditions too hot for the machines to continue to operate. The failure of HVAC systems for any reason is cause for concern.

Properly securing these systems is important in helping prevent an attacker from performing a physical DoS attack on your servers. Fire suppression systems should be specialized for the data center. Standard sprinkler-based systems are not optimal for data centers because water will ruin large electrical infrastructures and most integrated circuit–based devices—that is, computers. Gas-based systems are a good alternative, though they also carry special concerns. Halon was used for many years, and any existing installations may still have it for fire suppression in data centers. Halon displaces oxygen, and any people caught in the gas when the system goes off will need a breathing apparatus to survive. Halon is being replaced with other gas-based suppression systems, such as argon and nitrogen mixing systems or carbon dioxide, but the same danger to people exists, so these systems should be carefully implemented.

Authentication

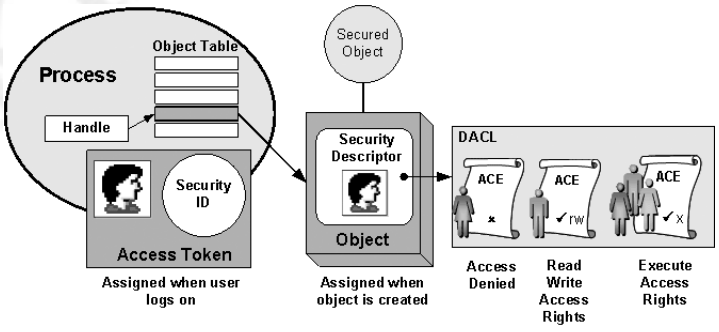
Authentication is the process by which a user proves that she is who she says she is. Authentication is performed to allow or deny a person access to a physical space. The heart of any access control system is to allow access to authorized users and to make sure access is denied to unauthorized people. Authentication is required because many companies have grown so large that not every employee knows every other employee, so

it can be difficult to tell by sight who is supposed to be where. Electronic access control systems were spawned from the need to have more logging and control than provided by the older method of metallic keys. Most electronic systems currently use a token-based card that if passed near a reader, and if you have permission from the system, will unlock the door strike and let you pass into the area. Newer technology attempts to make the authentication process easier and more secure.

Access Tokens

Access tokens are defined as "something you have." An access token is a physical object that identifies specific access rights, and in authentication falls into the "something you have" factor. Your house key, for example, is a basic physical access token that allows you access into your home. Although keys have been used to unlock devices for centuries, they do have several limitations. Keys are paired exclusively with a lock or a set of locks, and they are not easily changed. It is easy to add an authorized user by giving the user a copy of the key, but it is far more difficult to give that user selective access unless that specified area is already set up as a separate key. It is also difficult to take access away from a single key or key holder, which usually requires a rekey of the whole system.

The primary drawback of token-based authentication is that only the token is being authenticated. Therefore, the theft of the token could grant anyone who possessed the token access to what the system protects. The risk of theft of the token can be offset by the use of *multiple-factor authentication*. One of the ways that people have tried to achieve multiple-factor authentication is to add a biometric factor to the system.



Biometrics

Biometrics uses the measurements of certain biological factors to identify one specific person from others. These factors are based on parts of the human body that are unique. The most well-known of these unique biological factors is the fingerprint. However, many others can be used—for instance, the retina or iris of the eye, the geometry of the hand, and the geometry of the face. When these are used for authentication, there is a two part process, enrollment and then authentication. During enrollment, a computer takes the image of the biological factor and reduces it to a numeric value. When the user attempts to authenticate, this feature is scanned by the reader, and the computer compares the numeric value being read to the one stored in the database. If they match, access is allowed. Since these physical factors are unique, theoretically

only the actual authorized person would be allowed access.

A major concern with biometrics is that if someone is able to steal the uniqueness factor that the machine scans—your fingerprint from a glass, for example—and is able to reproduce that factor in a substance that fools the scanner, that person now has your access privileges. This idea is compounded by the fact that it is impossible for you to change your fingerprint if it gets stolen. It is easy to replace a lost or stolen token and delete the missing one from the system, but it is far more difficult to replace a human hand. Another problem with biometrics is that parts of the human body can change. A human face can change, through scarring, weight loss or gain, or surgery. A fingerprint can be changed through damage to the fingers. Eye retinas can be affected by some types of diabetes or pregnancy. All of these changes force the biometric system to allow a higher tolerance for variance in the biometric being read. This has led the way for high-security installations to move toward multiple-factor authentication.

Multiple-factor Authentication

Multiple-factor authentication is simply the combination of two or more types of authentication. Three broad categories of authentication can be used: what you are (for example, biometrics), what you have (for instance, tokens), and what you know (passwords and other information). Two-factor authentication combines any two of these before granting access. An example would be a card reader that then turns on a fingerprint scanner—if your fingerprint matches the one on file for the card, you are granted access. Three-factor authentication would combine all three types, such as a smart card reader that asks for a PIN before enabling a retina scanner. If all three correspond to a valid user in the computer database, access is granted.

Multiple-factor authentication methods greatly enhance security by making it very difficult for an attacker to obtain all the correct materials for authentication. They also protect against the risk of stolen tokens, as the attacker must have the correct biometric, password, or both. More important, it enhances the security of biometric systems. Multiple-factor authentication does this by protecting against a stolen biometric. Changing the token makes the biometric useless unless the attacker can steal the new token. It also reduces false positives by trying to match the supplied biometric with the one that is associated with the supplied token. This prevents the computer from seeking a match using the entire database of biometrics. Using multiple factors is one of the best ways to ensure proper authentication and access control.

Infrastructure Security

Infrastructure security begins with the design of the infrastructure itself. The proper use of components improves not only performance but security as well. Network components are not isolated from the computing environment and are an essential aspect of a total computing environment. From the routers, switches, and cables that connect the devices, to the firewalls and gateways that manage communication, from the network design to the protocols employed, all of these items play essential roles in both performance and security.

In the CIA of security, the A for availability is often overlooked. Yet it is availability that has moved computing into this networked framework, and this concept has played a significant role in security. A failure in security can easily lead to a failure in availability and hence a failure of the system to meet user needs.
Security failures can occur in two ways. First, a failure can allow unauthorized users access to resources and data they are not authorized to use, compromising information security. Second, a failure can prevent a user from accessing resources and data the user is authorized to use. This second failure is often overlooked, but it can be as serious as the first. The primary goal of network infrastructure security is to allow all authorized use and deny all unauthorized use of resources.

Devices
A complete network computer solution in today's business environment consists of more than just client computers and servers. Devices are needed to connect the clients and servers and to regulate the traffic between them. Devices are also needed to expand this network beyond simple client computers and servers to include yet other devices, such as wireless and handheld systems. Devices come in many forms and with many functions, from hubs and switches, to routers, wireless access points, and special-purpose devices such as virtual private network (VPN) devices. Each device has a specific network function and plays a role in maintaining network infrastructure security.

Workstations
Most users are familiar with the client computers used in the client/server model called *workstation* devices. The workstation is the machine that sits on the desktop and is used every day for sending and reading e-mail, creating spreadsheets, writing reports in a word processing program, and playing games. If a workstation is connected to a network, it is an important part of the security solution for the network. Many threats to information security can start at a workstation, but much can be done in a few simple steps to provide protection from many of these threats.

Workstations are attractive targets for crackers as they are numerous and can serve as entry points into the network and the data that is commonly the target of an attack. Although *safety* is a relative term, following these basic steps will increase workstation security immensely:

- Remove unnecessary protocols such as Telnet, NetBIOS, IPX.
- Remove modems unless needed and authorized.
- Remove all shares that are not necessary.
- Rename the administrator account, securing it with a strong password.
- Remove unnecessary user accounts.
- Install an antivirus program and keep abreast of updates.
- If the floppy drive is not needed, remove or disconnect it.
- Consider disabling USB ports via CMOS to restrict data movement to USB devices.
- If no corporate firewall exists between the machine and the Internet, install a firewall.
- Keep the operating system (OS) patched and up to date.

Antivirus Software for Workstations

Antivirus packages are available from a wide range of vendors. Running a network of computers without this basic level of protection will be an exercise in futility. Even though a virus attack is rare, the time and money you spend cleaning it up will more than equal the cost of antivirus protection. Even more important, once connected by networks, computers can spread a virus from machine to machine with an ease that's even greater than simple floppy disk transfer. One unprotected machine can lead to problems throughout a network as other machines have to use their antivirus software to attempt to clean up a spreading infection.

Even secure networks can fall prey to virus and worm contamination, and infection has been known to come from commercial packages. As important as antivirus software is, it is even more important to keep the virus definitions for the software up to date. Out-of-date definitions can lead to a false sense of security, and many of the most potent virus and worm attacks are the newest ones being developed. The risk associated with a new virus is actually higher than for many of the old ones, which have been eradicated to a great extent by antivirus software.

A virus is a piece of software that must be introduced to the network and then executed on a machine. Workstations are the primary mode of entry for a virus into a network. Although a lot of methods can be used to introduce a virus to a network, the two most common are transfer of an infected file from another networked machine and from e-mail. A lot of work has gone into software to clean e-mail while in transit and at the mail server. But transferred files are a different matter altogether. People bring files from home, from friends, from places unknown and then execute them on a PC for a variety of purposes. It doesn't matter whether it is a funny executable, a game, or even an authorized work application—the virus doesn't care what the original file is, it just uses it to gain access. Even sharing of legitimate work files and applications can introduce viruses.

Once considered by many users to be immune, Apple Macintosh computers had very few examples of malicious software in the wild. This was not due to anything other than a low market share, and hence the devices were ignored by the malware community as a whole. As Mac has increased in market share, so has its exposure, and today a variety of Mac OS X malware steals files and passwords and is even used to take users' pictures with the computer's built-in webcam. All user machines need to install antivirus software in today's environment, because any computer can become a target.

Additional Precautions for Workstations

Personal firewalls are a necessity if a machine has an unprotected interface to the Internet. These are seen less often in commercial networks, as it is more cost effective to connect through a firewall server. With the advent of broadband connections for homes and small offices, this needed device is frequently missed. This can result in penetration of a PC from an outside hacker or a worm infection. Worst of all, the workstation can become part of a larger attack against another network, unknowingly joining forces with other compromised machines in a distributed denial-of-service (DDoS) attack.

Servers

Servers are the computers in a network that host applications and data for everyone to share. Servers come in many sizes, from small single-CPU boxes that can be less powerful than a workstation, to multiple-CPU monsters, up to and including mainframes. The operating systems used by servers range from Windows Server, to Linux/UNIX, to Multiple Virtual Storage (MVS) and other mainframe operating systems. The OS on a server tends to be more robust than the OS on a workstation system and is designed to service multiple users over a network at the same time. Servers can host a variety of applications, including web servers, databases, e-mail servers, file servers, print servers, and application servers for middleware applications.

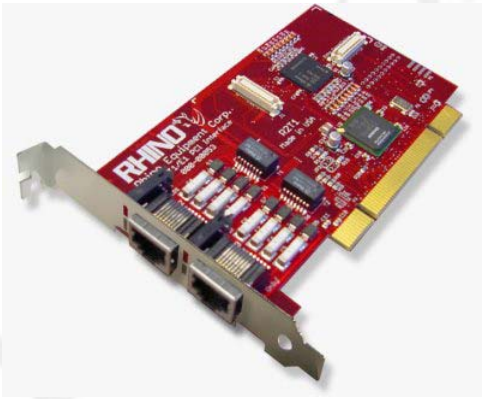
The key management issue behind running a secure server setup is to identify the specific needs of a server for its proper operation and enable only items necessary for those functions. Keeping all other services and users off the system improves system throughput and increases security. Reducing the attack surface area associated with a server reduces the vulnerabilities now and in the future as updates are required.

Once a server has been built and is ready to place into operation, the recording of MD5 hash values on all of its crucial files will provide valuable information later in case of a question concerning possible system integrity after a detected intrusion. The use of hash values to detect changes was first developed by Gene Kim and Eugene Spafford at Purdue University in 1992. The concept became the product Tripwire, which is now available in commercial and open source forms. The same basic concept is used by many security packages to detect file level changes.

Antivirus Software for Servers

The need for antivirus protection on servers depends a great deal on the use of the server. Some types of servers, such as e-mail servers, can require extensive antivirus protection because of the services they provide. Other servers (domain controllers and remote access servers, for example) may not require any antivirus software, as they do not allow users to place files on them. File servers will need protection, as will certain types of application servers. There is no general rule, so each server and its role in the network will need to be examined for applicability of antivirus software.

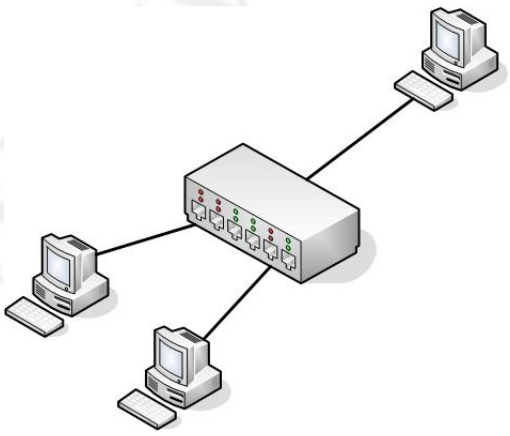
Network Interface Cards



To connect a server or workstation to a network, a device known as a *network interface card (NIC)* is used. A NIC is a card with a connector port for a particular type of network connection, either Ethernet or Token Ring. The most common network type in use for local area networks is the Ethernet protocol, and the most common connector is the RJ-45 connector. Figure 8-1 shows a RJ-45 connector (lower) compared to a standard telephone connector (upper). Additional types of connectors include coaxial cable connectors, frequently used with cable modems and extending from the wall to the cable modem.

The purpose of a NIC is to provide lower level protocol functionality from the OSI (Open System Interconnection) model. A NIC is the physical connection between a computer and the network. As the NIC defines the type of physical layer connection, different NICs are used for different physical protocols. NICs come as single-port and multiport, and most workstations use only a single-port NIC, as only a single network connection is needed. For servers, multiport NICs are used to increase the number of network connections, increasing the data throughput to and from the network.

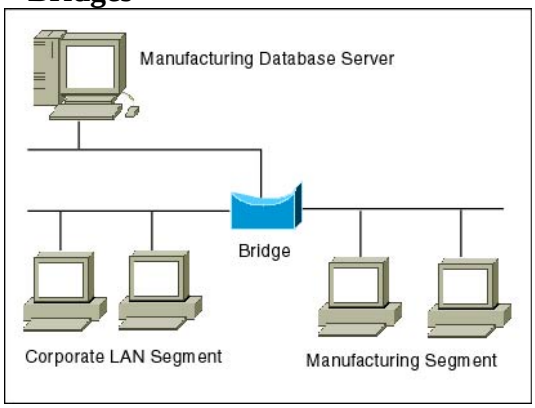
Hubs



Hubs are networking equipment that connects devices using the same protocol at the physical layer of the OSI model. A hub allows multiple machines in an area to be connected together in a star configuration with the hub as the center. This configuration can save significant amounts of cable and is an efficient method of configuring an Ethernet backbone. All connections on a hub share a single *collision domain*, a small cluster in a network where collisions occur. As network traffic increases, it can become limited by collisions. The collision issue has made hubs obsolete in newer, higher

performance networks, with low-cost switches and switched Ethernet keeping costs low and usable bandwidth high. Hubs also create a security weakness in that all connected devices see all traffic, enabling sniffing and eavesdropping to occur.

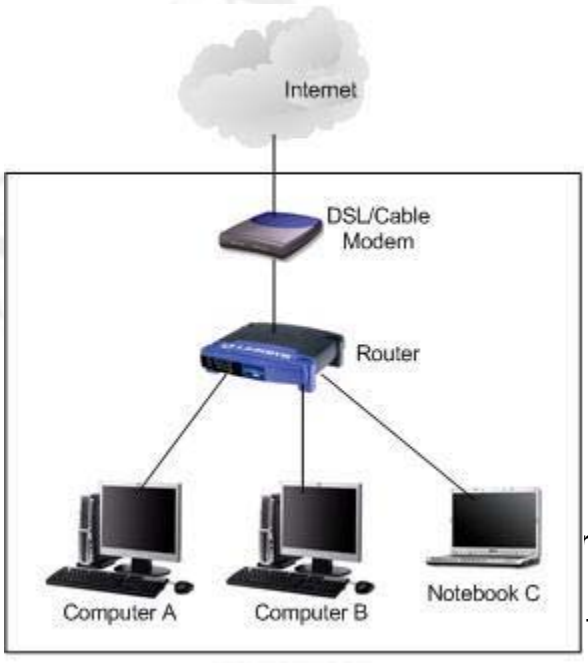
Bridges



Bridges are networking equipment that connects devices using the same protocol at the physical layer of the OSI model. A bridge operates at the data link layer, filtering traffic based on MAC addresses. Bridges can reduce collisions by separating pieces of a network into two separate collision domains, but this only cuts the collision problem in half. Although bridges are useful, a better solution is to use switches for network connections.

Switches

Switches form the basis for connections in most Ethernet-based local area networks (LANs). Although hubs and bridges still exist, in today's high-performance network environment switches have replaced both. A switch has separate collision domains for each port. This means that for each port, two collision domains exist: one from the port to the client on the downstream side and one from the switch to the network upstream.



When *full duplex* is employed, collisions are virtually eliminated from the two nodes, host and client. This also acts as a security factor in that a sniffer can see only limited traffic, as opposed to a hub-based system, where a single sniffer can see all of the traffic to and from connected devices.

Switches operate at the data link layer, while routers act at the network layer. For intranets, switches have become what routers are on the Internet—the device of choice for connecting machines. As

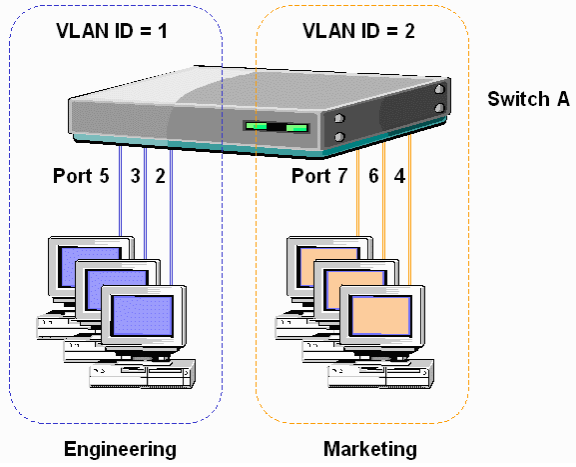
switches have become the primary network connectivity device, additional functionality has been added to them. A switch is usually a layer 2 device, but layer 3 switches incorporate routing functionality.

Switches can also perform a variety of security functions. Switches work by moving packets from inbound connections to outbound connections. While moving the packets, it is possible to inspect the packet headers and enforce security policies. Port address security based on MAC addresses can determine whether a packet is allowed or blocked from a connection. This is the very function that a firewall uses for its determination, and this same functionality is what allows an 802.1x device to act as an "edge device."

Virtual Local Area Networks

The other security feature that can be enabled in some switches is the concept of *virtual local area networks (VLANs)*. Cisco defines a VLAN as a "broadcast domain within a switched network," meaning that information is carried in broadcast mode only to devices within a VLAN. Switches that allow multiple VLANs to be defined enable broadcast messages to be segregated into the specific VLANs. If each floor of an office, for example, were to have a single switch and you had accounting functions on two floors, engineering functions on two floors, and sales functions on two floors, then separate VLANs for accounting, engineering, and sales would allow separate broadcast domains for each of these groups, even those that spanned floors. This configuration increases network segregation, increasing throughput and security.

Unused switch ports can be preconfigured into empty VLANs that do not connect to the rest of the network. This significantly increases security against unauthorized network connections. If, for example, a building is wired with network connections in all rooms, including multiple connections for convenience and future expansion, these unused ports become open to the network. One solution is to disconnect the connection at the switch, but this merely moves the network opening into the switch room.



The better solution is to disconnect it and disable the port in the switch. This can be accomplished by connecting all unused ports into a VLAN that isolates them from the rest of the network.

Routers

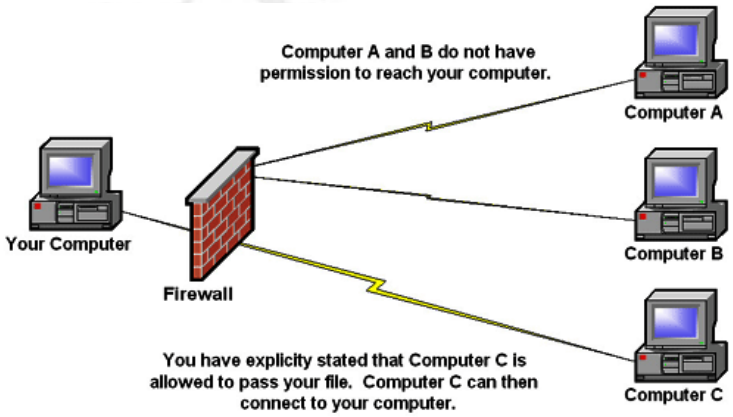
Routers are network traffic management devices used to connect different network segments together.

Routers operate at the network layer of the OSI model, routing traffic using the network address (typically an IP address) utilizing routing protocols to determine optimal routing paths across a network. Routers form the backbone of the Internet, moving traffic from network to network, inspecting packets from every communication as they move traffic in optimal paths.

Routers operate by examining each packet, looking at the destination address, and using algorithms and tables to determine where to send the packet next. This process of examining the header to determine the next hop can be done in quick fashion. Routers use access control lists (ACLs) as a method of deciding whether a packet is allowed to enter the network. With ACLs, it is also possible to examine the source address and determine whether or not to allow a packet to pass. This allows routers equipped with ACLs to drop packets according to rules built in the ACLs. This can be a cumbersome process to set up and maintain, and as the ACL grows in size, routing efficiency can be decreased. It is also possible to configure some routers to act as quasi–application gateways, performing stateful packet inspection and using contents as well as IP addresses to determine whether or not to permit a packet to pass. This can tremendously increase the time for a router to pass traffic and can significantly decrease router throughput.

Firewalls

A *firewall* can be hardware, software, or a combination whose purpose is to enforce a set of network security policies across network connections. It is much like a wall with a window: the wall serves to keep things out, except those permitted through the window. Network security policies act like the glass in the window; they permit some things to pass, such as light, while blocking others, such as air. The heart of a firewall is the set of security policies that it enforces. Management determines what is allowed in the form of network traffic between devices, and these policies are used to build rule sets for the firewall devices used to filter network traffic across the network.



Security policies are rules that define what traffic is permissible and what traffic is to be blocked or denied. These are not universal rules, and many different sets of rules are created for a single company with multiple connections. A web server connected to the Internet may be configured to allow traffic only on port 80 for HTTP and have all other ports blocked, for example. An e-mail server may have only necessary ports for e-mail open, with others blocked. The network firewall can be programmed to block all traffic to the web server except for port 25. In this fashion, the firewall acts as a security filter, enabling control

over network traffic, by machine, by port, and in some cases based on application level detail. A key to setting security policies for firewalls is the same as has been seen for other security policies—the principle of least access. Allow only the necessary access for a function; block or deny all unneeded functionality. How a firm deploys its firewalls determines what is needed for security policies for each firewall.

How Do Firewalls Work?
Firewalls enforce the established security policies through a variety of mechanisms, including the following:

- Network Address Translation (NAT)
- Basic packet filtering
- Stateful packet filtering
- ACLs
- Application layer proxies

One of the most basic security functions provided by a firewall is NAT, which allows you to mask significant amounts of information from outside of the network. This allows an outside entity to communicate with an entity inside the firewall without truly knowing its address. NAT is a technique used in IPv4 to link private IP addresses to public ones. Private IP addresses are sets of IP addresses that can be used by anyone and by definition are not routable across the Internet. NAT can assist in security by preventing direct access to devices from outside the firm, without first having the address changed at a NAT device. The benefit is less public IP addresses are needed, and from a security point of view the internal address structure is not known to the outside world. If a hacker attacks the source address, he is simply attacking the NAT device, not the actual sender of the packet.

NAT was conceived to resolve an address shortage associated with IPv4 and is considered by many to be unnecessary for IPv6. The added security features of enforcing traffic translation and hiding internal network details from direct outside connections will give NAT life well into the IPv6 timeframe.

Basic packet filtering, the next most common firewall technique, involves looking at packets, their ports, protocols, source and destination addresses, and checking that information against the rules configured on the firewall. Telnet and FTP connections may be prohibited from being established to a mail or database server, but they may be allowed for the respective service servers. This is a fairly simple method of filtering based on information in each packet header, such as IP addresses and TCP/UDP ports. Packet filtering will not detect and catch all undesired packets, but it is fast and efficient.

Wireless
Wireless devices bring additional security concerns. There is, by definition, no physical connection to a wireless device; radio waves or infrared carry data, which allows anyone within range access to the data. This means that unless you take specific precautions, you

have no control over who can see your data. Placing a wireless device behind a firewall does not do any good, because the firewall stops only physically connected traffic from reaching the device. Outside traffic can come literally from the parking lot directly to the wireless device.

The point of entry from a wireless device to a wired network is performed at a device called a *wireless access point.* Wireless access points can support multiple concurrent devices accessing network resources through the network node they provide.

Several mechanisms can be used to add wireless functionality to a machine. For PCs, this can be done via an expansion card. For notebooks, a PCMCIA adapter for wireless networks is available from several vendors. For both PCs and notebooks, vendors have introduced USB-based wireless connectors.

Modems

Modems were once a slow method of remote connection that was used to connect client workstations to remote services over standard telephone lines. *Modem* is a shortened form of *modulator/demodulator*, covering the functions actually performed by the device as it converts analog signals to digital and vice versa. To connect a digital computer signal to the analog telephone line required one of these devices. Today, the use of the term has expanded to cover devices connected to special digital telephone lines—DSL modems—and to cable television lines—cable modems. Although these devices are not actually modems in the true sense of the word, the term has stuck through marketing efforts directed to consumers. DSL and cable modems offer broadband high-speed connections and the opportunity for continuous connections to the Internet. Along with these new desirable characteristics come some undesirable ones, however. Although they both provide the same type of service, cable and DSL modems have some differences. A DSL modem provides a direct connection between a subscriber's computer and an Internet connection at the local telephone company's switching station.

This private connection offers a degree of security, as it does not involve others sharing the circuit. Cable modems are set up in shared arrangements that theoretically could allow a neighbor to sniff a user's cable modem traffic. Both cable and DSL services are designed for a continuous connection, which brings up the question of IP address life for a client. Although some services originally used a static IP arrangement, virtually all have now adopted the Dynamic Host Configuration Protocol (DHCP) to manage their address space. A static IP has an advantage of being the same and enabling convenient DNS connections for outside users. As cable and DSL services are primarily designed for client services as opposed to host services, this is not a relevant issue. A

security issue of a static IP is that it is a stationary target for hackers. The move to DHCP has not significantly lessened this threat, however, for the typical IP lease on a cable modem DHCP is for days. This is still relatively stationary, and some form of firewall protection needs to be employed by the user.

Cable/DSL Security
The modem equipment provided by the subscription service converts the cable or DSL signal into a standard Ethernet signal that can then be connected to a NIC on the client device. This is still just a direct network connection, with no security device separating the two. The most common security device used in cable/DSL connections is a firewall. The firewall needs to be installed between the cable/DSL modem and client computers.

Telecom/PBX
Private branch exchanges (PBXs) are an extension of the public telephone network into a business. Although typically considered a separate entity from data systems, they are frequently interconnected and have security requirements as part of this interconnection as well as of their own. PBXs are computer-based switching equipment designed to connect telephones into the local phone system. Basically digital switching systems, they can be compromised from the outside and used by phone hackers (phreakers) to make phone calls at the business' expense. Although this type of hacking has decreased with lower cost long distance, it has not gone away, and as several firms learn every year, voice mail boxes and PBXs can be compromised and the long-distance bills can get very high, very fast.

Another problem with PBXs arises when they are interconnected to the data systems, either by corporate connection or by rogue modems in the hands of users. In either case, a path exists for connection to outside data networks and the Internet. Just as a firewall is needed for security on data connections, one is needed for these connections as well. Telecommunications firewalls are a distinct type of firewall designed to protect both the PBX and the data connections. The functionality of a telecommunications firewall is the same as that of a data firewall: it is there to enforce security policies.

Telecommunication security policies can be enforced even to cover hours of phone use to prevent unauthorized long-distance usage through the implementation of access codes and/or restricted service hours.

RAS
Remote Access Service (RAS) is a portion of the Windows OS that allows the connection between a client and a server via a dial-up telephone connection. Although slower than cable/DSL connections, this is still a common method for connecting to a remote network. When a user dials into the computer system, authentication and authorization are performed through a series of remote access protocols. For even greater security, a callback system can be employed, where the server calls back to the client at a set telephone number for the data exchange. RAS can also mean Remote Access Server, a

term for a server designed to permit remote users access to a network and to regulate their access. A variety of protocols and methods exist to perform this function.

VPN
A virtual private network (VPN) is a construct used to provide a secure communication channel between users across public networks such as the Internet. A variety of techniques can be employed to instantiate a VPN connection.

The use of encryption technologies allows either the data in a packet to be encrypted or the entire packet to be encrypted. If the data is encrypted, the packet header can still be sniffed and observed between source and destination, but the encryption protects the contents of the packet from inspection. If the entire packet is encrypted, it is then placed into another packet and sent via tunnel across the public network. Tunneling can protect even the identity of the communicating parties.

The most common implementation of VPN is via IPsec, a protocol for IP security. IPsec is mandated in IPv6 and is optionally back-fitted into IPv4. IPsec can be implemented in hardware, software, or a combination of both.

Intrusion Detection Systems
Intrusion detection systems (IDSs) are designed to detect, log, and respond to unauthorized network or host use, both in real time and after the fact. IDSs are available from a wide selection of vendors and are an essential part of network security. These systems are implemented in software, but in large systems, dedicated hardware is required as well. IDSs can be divided into two categories: network-based systems and host-based systems. Two primary methods of detection are used: signature-based and anomaly-based.

Network Access Control
Networks comprise connected workstations and servers. Managing security on a network involves managing a wide range of issues, from various connected hardware and the software operating these devices. Assuming that the network is secure, each additional connection involves risk. Managing the endpoints on a case-by-case basis as they connect is a security methodology known as *network access control.* Two main competing methodologies exist: Network Access Protection (NAP) is a Microsoft technology for controlling network access of a computer host, and Network Admission Control (NAC) is Cisco's technology for controlling network admission.

Both the Cisco NAC and Microsoft NAP are in their early stages of implementation. The concept of automated admission checking based on client device characteristics is here to stay, as it provides timely control in the ever-changing network world of today's enterprises.

Network Monitoring/Diagnostic

The computer network itself can be considered a large computer system, with performance and operating issues. Just as a computer needs management, monitoring, and fault resolution, so do networks. SNMP was developed to perform this function across networks. The idea is to enable a central monitoring and control center to maintain, configure, and repair network devices, such as switches and routers, as well as other network services such as firewalls, IDSs, and remote access servers. SNMP has some security limitations, and many vendors have developed software solutions that sit on top of SNMP to provide better security and better management tool suites.

The concept of a network operations center (NOC) comes from the old phone company network days, when central monitoring centers monitored the health of the telephone network and provided interfaces for maintenance and management. This same concept works well with computer networks, and companies with midsize and larger networks employ the same philosophy. The NOC allows operators to observe and interact with the network, using the self-reporting and in some cases self-healing nature of network devices to ensure efficient network operation. Although generally a boring operation under normal conditions, when things start to go wrong, as in the case of a virus or worm attack, the center can become a busy and stressful place as operators attempt to return the system to full efficiency while not interrupting existing traffic.

As networks can be spread out literally around the world, it is not feasible to have a person visit each device for control functions. Software enables controllers at NOCs to measure the actual performance of network devices and make changes to the configuration and operation of devices remotely. The ability to make remote connections with this level of functionality is both a blessing and a security issue. Although this allows efficient network operations management, it also provides an opportunity for unauthorized entry into a network. For this reason, a variety of security controls are used, from secondary networks to VPNs and advanced authentication methods with respect to network control connections.

Mobile Devices

Mobile devices such as personal digital assistants (PDAs) and mobile phones are the latest devices to join the corporate network. These devices can perform significant business functions, and in the future, more of them will enter the corporate network and more work will be performed with them. These devices add several challenges for network administrators. When they synchronize their data with that on a workstation or server, the opportunity exists for viruses and malicious code to be introduced to the network. This can be a major security gap, as a user may access separate e-mail accounts, one personal, without antivirus protection, the other corporate. Whenever data is moved from one network to another via the PDA, the opportunity to load a virus onto the workstation exists. Although the virus may not affect the PDA or phone, these devices can act as transmission vectors. Currently, at least one vendor offers antivirus protection for PDAs, and similar protection for phones is not far away.

Media

The base of communications between devices is the physical layer of the OSI model. This is the domain of the actual connection between devices, whether by wire, fiber, or radio frequency waves. The physical layer separates the definitions and protocols required to transmit the signal physically between boxes from higher level protocols that deal with the details of the data itself. Four common methods are used to connect equipment at the physical layer:

- Coaxial cable
- Twisted-pair cable
- Fiber-optics
- Wireless

Coaxial Cable

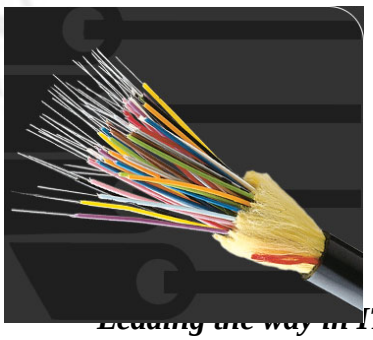
Coaxial cable is familiar to many households as a method of connecting televisions to VCRs or to satellite or cable services. It is used because of its high bandwidth and shielding capabilities. Compared to standard twisted-pair lines such as telephone lines, "coax" is much less prone to outside interference. It is also much more expensive to run, both from a cost-per-foot measure and from a cable-dimension measure. Coax costs much more per foot than standard twisted pair and carries only a single circuit for a large wire diameter.

UTP/STP

Twisted-pair wires have all but completely replaced coaxial cables in Ethernet networks. Twisted-pair wires use the same technology used by the phone company for the movement of electrical signals. Single pairs of twisted wires reduce electrical crosstalk and electromagnetic interference. Multiple groups of twisted pairs can then be bundled together in common groups and easily wired between devices.

Twisted pairs come in two types, shielded and unshielded. Shielded twisted-pair (STP) has a foil shield around the pairs to provide extra shielding from electromagnetic interference. Unshielded twisted-pair (UTP) relies on the twist to eliminate interference. UTP has a cost advantage over STP and is usually sufficient for connections, except in very noisy electrical areas.

Fiber



Fiber-optic cable uses beams of laser light to connect devices over a thin glass wire. The biggest advantage to fiber is its bandwidth, with transmission capabilities into the terabits per second range. Fiber-optic cable is used to make high-speed connections between servers and is the backbone medium of the Internet and large networks. For all of its speed and bandwidth advantages, fiber have one major drawback—cost. The cost of using fiber is a two-

edged sword. It is cheaper when measured by bandwidth to use fiber than competing wired technologies. The length of runs of fiber can be much longer, and the data capacity of fiber is much higher. But connections to a fiber are difficult and expensive and fiber is impossible to splice. Making the precise connection on the end of a fiber-optic line is a highly skilled job and is done by specially trained professionals who maintain a level of proficiency. Once the connector is fitted on the end, several forms of connectors and blocks are used.

Unguided Media

Electromagnetic waves have been transmitted to convey signals literally since the inception of radio. *Unguided media* is a phrase used to cover all transmission media not guided by wire, fiber, or other constraints; it includes radio frequency (RF), infrared (IR), and microwave methods. Unguided media have one attribute in common: they are unguided and as such can travel to many machines simultaneously. Transmission patterns can be modulated by antennas, but the target machine can be one of many in a reception zone. As such, security principles are even more critical, as they must assume that unauthorized users have access to the signal.

Infrared

Infrared (IR) is a band of electromagnetic energy just beyond the red end of the visible color spectrum. IR has been used in remote control devices for years, and it cannot penetrate walls but instead bounces off them. IR made its debut in computer networking as a wireless method to connect to printers. Now that wireless keyboards, wireless mice, and PDAs exchange data via IR, it seems to be everywhere. IR can also be used to connect devices in a network configuration, but it is slow compared to other wireless technologies. It also suffers from not being able to penetrate solid objects, so stack a few items in front of the transceiver and the signal is lost.

RF/Microwave

The use of radio frequency (RF) waves to carry communication signals goes back to the beginning of the twentieth century. RF waves are a common method of communicating in a wireless world. They use a variety of frequency bands, each with special characteristics. The term *microwave* is used to describe a specific portion of the RF spectrum that is used for communication as well as other tasks, such as cooking. Point-to-point microwave links have been installed by many network providers to carry communications over long distances and rough terrain. Microwave communications of telephone conversations were the basis for forming the telecommunication company MCI. Many different frequencies are used in the microwave bands for many different purposes. Today, home users can use wireless networking throughout their house and enable laptops to surf the Web while they move around the house. Corporate users are experiencing the same phenomenon, with wireless networking enabling corporate users to check e-mail on laptops while riding a shuttle bus on a business campus

Security Concerns for Transmission Media

The primary security concern for a system administrator has to be preventing physical access to a server by an unauthorized individual. Such access will almost always spell disaster, for with direct access and the correct tools, any system can be infiltrated. One of the administrator's next major concerns should be preventing unfettered access to a network connection. Access to switches and routers is almost as bad as direct access to a server, and access to network connections would rank third in terms of worst-case scenarios. Preventing such access is costly, yet the cost of replacing a server because of theft is also costly.

Physical Security

A balanced approach is the most sensible approach when addressing physical security, and this applies to transmission media as well. Keeping network switch rooms secure and cable runs secure seems obvious, but cases of using janitorial closets for this vital business purpose abound. One of the keys to mounting a successful attack on a network is information. Usernames, passwords, server locations—all of these can be obtained if someone has the ability to observe network traffic in a process called *sniffing*. A sniffer can record all the network traffic and this data can be mined for accounts, passwords, and traffic content, all of which can be useful to an unauthorized user. Many common scenarios exist when unauthorized entry to a network occurs, including these:

- Inserting a node and functionality that is not authorized on the network, such as a sniffer device or unauthorized wireless access point
- Modifying firewall security policies
- Modifying ACLs for firewalls, switches, or routers
- Modifying network devices to echo traffic to an external node

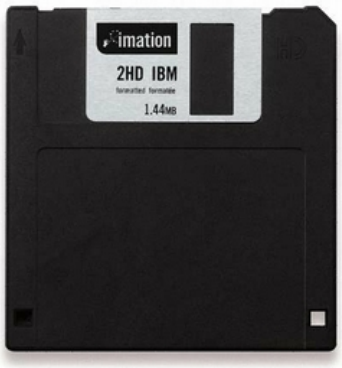
One starting point for many intrusions is the insertion of an unauthorized sniffer into the network, with the fruits of its labors driving the remaining unauthorized activities. The best first effort is to secure the actual network equipment to prevent this type of intrusion.

Wireless networks make the intruder's task even easier, as they take the network to the users, authorized or not. A technique called *war-driving* involves using a laptop and software to find wireless networks from outside the premises. A typical use of war driving is to locate a wireless network with poor (or no) security and obtain free Internet access, but other uses can be more devastating. Methods for securing even the relatively weak Wired Equivalent Privacy (WEP) protocol are not difficult; they are just typically not followed. A simple solution is to place a firewall between the wireless access point and the rest of the network and authenticate users before allowing entry.

Home users can do the same thing to prevent neighbors from "sharing" their Internet connections. To ensure that unauthorized traffic does not enter your network through a wireless access point, you must either use a firewall with an authentication system or establish a VPN.

Removable Media

One concept common to all computer users is data storage. Sometimes storage occurs on a file server and sometimes on movable media, allowing it to be transported between machines. Moving storage media represents a security risk from a couple of angles, the first being the potential loss of control over the data on the moving media.



Second is the risk of introducing unwanted items, such as a virus or a worm, when the media are attached back to a network. Both of these issues can be remedied through policies and software. The key is to ensure that they are occurring. To describe media-specific issues, the media can be divided into three categories: magnetic, optical, and electronic.

Magnetic Media

Magnetic media store data through the rearrangement of magnetic particles on a nonmagnetic substrate. Common forms include hard drives, floppy disks, zip disks, and magnetic tape. Although the specific format can differ, the basic concept is the same. All these devices share some common characteristics: Each has sensitivity to external magnetic fields. Attach a floppy disk to the refrigerator door with a magnet if you want to test the sensitivity. They are also affected by high temperatures as in fires and by exposure to water.

Hard Drives

Hard drives used to require large machines in mainframes. Now they are small enough to attach to PDAs and handheld devices. The concepts remain the same among all of them: a spinning platter rotates the magnetic media beneath heads that read the patterns in the oxide coating. As drives have gotten smaller and rotation speeds increased, the capacities have also grown. Today gigabytes can be stored in a device slightly larger than a bottle cap. Portable hard drives in the 120 to 320GB range are now available and affordable.

One of the latest advances is full drive encryption built into the drive hardware. Using a key that is controlled, through a Trusted Platform Module (TPM) interface for instance, this technology protects the data if the drive itself is lost or stolen. This may not be important if a thief takes the whole PC, but in larger storage environments, drives are placed in separate boxes and remotely accessed. In the specific case of notebook machines, this layer can be tied to smart card interfaces to provide more security. As this is built into the controller, encryption

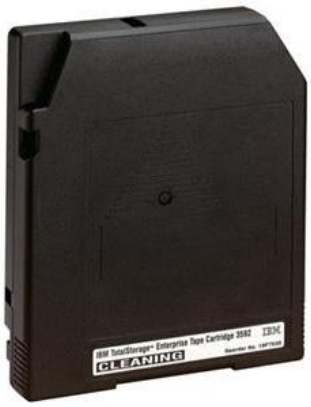
protocols such as Advanced Encryption Standard (AES) and Triple Data Encryption Standard (3DES) can be performed at full drive speed.

Diskettes

Floppy disks were the computer industry's first attempt at portable magnetic media. The movable medium was placed in a protective sleeve, and the drive remained in the machine. Capacities up to 1.4MB were achieved, but the fragility of the device as the size increased, as well as competing media, has rendered floppies almost obsolete. A better alternative, the Zip disk from Iomega Corporation, improved on the floppy with a stronger case and higher capacity (250MB); it has been a common backup and file transfer medium. But even the increased size of 250MB is not large enough for some multimedia files, and recordable optical (CD-R) drives have arrived to fill the gap; they will be discussed shortly.

Tape

Magnetic tape has held a place in computer centers since the beginning of computing. Their primary use has been bulk offline storage and backup. Tape functions well in this role because of its low cost. The disadvantage of tape is its nature as a serial access medium, making it slow to work with for large quantities of data. Several types of magnetic tape are in use today, ranging from quarter inch to digital linear tape (DLT) and digital audio tape (DAT). These cartridges can hold upward of 60GB of compressed data.

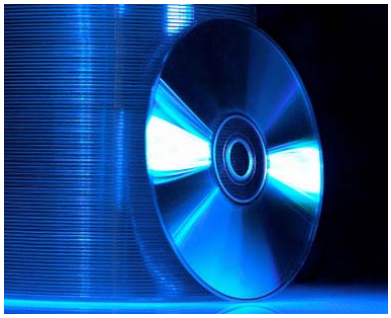


Tapes are still a major concern from a security perspective, as they are used to back up many types of computer systems. The physical protection afforded the tapes is of concern, because if a tape is stolen, an unauthorized user could establish a network and recover your data on his system, because it's all stored on the tape. Offsite storage is needed for proper disaster recovery protection, but secure offsite storage and transport is what is really needed. This important issue is frequently overlooked in many facilities. The simple solution to maintain control over the data even when you can't control the tape is through encryption. Backup utilities can secure the backups with encryption, but this option is frequently not used for a variety of reasons. Regardless of the rationale for not encrypting data, once a tape is lost, not using the encryption option becomes a lamented decision.

Optical Media

Optical media involve the use of a laser to read data stored on a physical device. Rather than a magnetic head picking up magnetic marks on a disk, a laser picks up deformities embedded in the media that contain the information. As with magnetic media, optical media can be read-write, although the read-only version is still more common.

CD-R/DVD



The compact disc (CD) took the music industry by storm, and then it took the computer industry by storm as well. A standard CD holds more than 640MB of data, in some cases up to 800 MB. The digital video disc (DVD) can hold almost 4GB of data. These devices operate as optical storage, with little marks burned in them to represent 1's and 0's on a microscopic scale. The most common type of CD is the read-only version, in which the data is written to the disc once and only read afterward. This has become a popular method for distributing computer software, although higher capacity DVDs have begun to replace CDs for program distribution.

DVDs will eventually occupy the same role that CDs have in the recent past, except that they hold more than seven times the data of a CD. This makes full-length movie recording possible on a single disc. The increased capacity comes from finer tolerances and the fact that DVDs can hold data on both sides. A wide range of formats for DVDs include DVD+R, DVD-R, dual layer, and now HD formats, HD-DVD and Blu-ray. This variety is due to competing "standards" and can result in confusion. DVD+R and -R are distinguishable only when recording, and most devices since 2004 should read both. Dual layers add additional space but require appropriate dual-layer–enabled drives.

HD-DVD and Blue-ray are competing formats in the high-definition arena, with devices that currently hold 50GB and with research prototypes promising up to 1TB on a disk. In 2008, Toshiba, the leader of the HD-DVD format, announced it was ceasing production, casting doubts onto its future, although this format is also used in gaming systems such as the Xbox 360.

Electronic Media

The latest form of removable media is electronic memory. Electronic circuits of static memory, which can retain data even without power, fill a niche where high density and small size are needed. Originally used in audio devices and digital cameras, these electronic media come in a variety of vendor-specific types, such as smart cards, SmartMedia, flash cards, memory sticks, and CompactFlash devices. Several recent photo-quality color printers have been released with ports to accept the cards directly, meaning that a computer is not required for printing. Computer readers are also available to permit storing data from the card onto hard drives and other media in a computer. The size of storage on these devices ranges from 256MB to 32GB and higher.

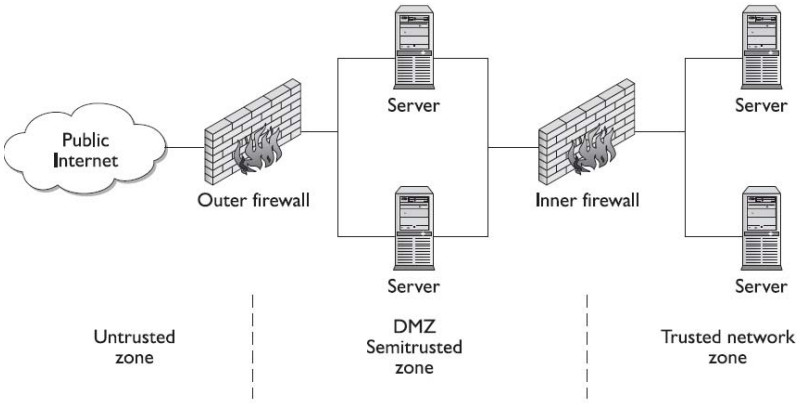
The advent of large capacity USB sticks has enabled users to build entire systems, OSs, and tools onto them to ensure security and veracity of the OS and tools. With the expanding use of virtualization, a user could carry an entire system on a USB stick and boot it using virtually any hardware. The only downside to this form of mobile computing is the slower speed of the USB 2.0 interface, currently limited to 480 Mbps.

Security Topologies
Networks are different than single servers; networks exist as connections of multiple devices. A key characteristic of a network is its layout, or *topology*. A proper network topology takes security into consideration and assists in "building security" into the network. Security-related topologies include separating portions of the network by use and function, strategically designing in points to monitor for IDS systems, building in redundancy, and adding fault-tolerant aspects.

Security Zones
The first aspect of security is a layered defense. Just as a castle has a moat, an outside wall, an inside wall, and even a keep, so, too, does a modern secure network have different layers of protection. Different zones are designed to provide layers of defense, with the outermost layers providing basic protection and the innermost layers providing the highest level of protection. A constant issue is that accessibility tends to be inversely related to level of protection, so it is more difficult to provide complete protection and unfettered access at the same time. Trade-offs between access and security are handled through zones, with successive zones guarded by firewalls enforcing ever-increasingly strict security policies. The outermost zone is the Internet, a free area, beyond any specific controls. Between the inner secure corporate network and the Internet is an area where machines are considered at risk. This zone has come to be called the *DMZ*, after its military counterpart, the demilitarized zone, where neither side has any specific controls. Once inside the inner secure network, separate branches are frequently carved out to provide specific functionality; under this heading, we will discuss intranets, extranets, and virtual LANs (VLANs).

DMZ
The DMZ is a military term for ground separating two opposing forces, by agreement and for the purpose of acting as a buffer between the two sides. A DMZ in a computer network is used in the same way; it acts as a buffer zone between the Internet, where no controls exist, and the inner secure network, where an organization has security policies in place (see Figure 8-4). To demarcate the zones and enforce separation, a firewall is used on each side of the DMZ. The area between these firewalls is accessible from either the inner secure network or the Internet. Figure 8-4 illustrates these zones as caused by firewall placement. The firewalls are specifically designed to prevent access across the DMZ directly, from the Internet to the inner secure network. Special attention should be paid to the security settings of network devices placed in the DMZ, and they should be considered at all times to be compromised by unauthorized use. A common industry term, *hardened operating system,* applies to machines whose functionality is locked down to preserve security. This approach needs to be applied to the machines in the

DMZ, and although it means that their functionality is limited, such precautions ensure that the machines will work properly in a less-secure environment.



The idea behind the use of the DMZ topology is to force an outside user to make at least one hop in the DMZ before he can access information inside the trusted network. If the outside user makes a request for a resource from the trusted network, such as a data element from a database via a web page, then this request needs to follow this scenario:

1. A user from the untrusted network (the Internet) requests data via a web page from a web server in the DMZ.
2. The web server in the DMZ requests the data from the application server, which can be in the DMZ or in the inner trusted network.
3. The application server requests the data from the database server in the trusted network.
4. The database server returns the data to the requesting application server.
5. The application server returns the data to the requesting web server.
6. The web server returns the data to the requesting user from the untrusted network.

This separation accomplishes two specific, independent tasks. First, the user is separated from the request for data on a secure network. By having intermediaries do the requesting, this layered approach allows significant security levels to be enforced. Users do not have direct access or control over their requests, and this filtering process can put controls in place. Second, scalability is more easily realized. The multiple-server solution can be made to be very scalable literally to millions of users, without slowing down any particular layer.

Internet

The Internet is a worldwide connection of networks and is used to transport e-mail, files, financial records, remote access—you name it—from one network to another. The Internet is not as a single network, but a series of interconnected networks that allow protocols to operate to enable data to flow across it. This means that even if your network doesn't have direct contact with a resource, as long as a neighbor, or a neighbor's neighbor, and so on, can get there, so can you. This large web allows users almost infinite ability to communicate between systems.

Because everything and everyone can access this interconnected web and it is outside of your control and ability to enforce security policies, the Internet should be considered an untrusted network. A firewall should exist at any connection between your trusted network and the Internet. This is not to imply that the Internet is a bad thing—it is a great resource for all networks and adds significant functionality to our computing environments.

The term World Wide Web (WWW) is frequently used synonymously to represent the Internet, but the WWW is actually just one set of services available via the Internet. WWW is more specifically the Hypertext Transfer Protocol (HTTP)–based services that are made available over the Internet. This can include a variety of actual services and content, including text files, pictures, streaming audio and video, and even viruses and worms.

Intranet

Intranet is a term used to describe a network that has the same functionality as the Internet for users but lies completely inside the trusted area of a network and is under the security control of the system and network administrators. Typically referred to as *campus* or *corporate* networks, intranets are used every day in companies around the world. An intranet allows a developer and a user the full set of protocols—HTTP, FTP, instant messaging, and so on—that is offered on the Internet, but with the added advantage of trust from the network security. Content on intranet web servers is not available over the Internet to untrusted users. This layer of security offers a significant amount of control and regulation, allowing users to fulfill business functionality while ensuring security.

Should users inside the intranet require access to information from the Internet; a proxy server can be used to mask the requestor's location. This helps secure the intranet from outside mapping of its actual topology. All Internet requests go to the proxy server. If a request passes filtering requirements, the proxy server, assuming it is also a cache server, looks in its local cache of previously downloaded web pages. If it finds the page in its cache, it returns the page to the requestor without needing to send the request to the Internet. If the page is not in the cache, the proxy server, acting as a client on behalf of the user, uses one of its own IP addresses to request the page from the Internet. When the page is returned, the proxy server relates it to the original request and forwards it on to the user. This masks the user's IP address from the Internet. Proxy servers can perform several functions for a firm; for example, they can monitor traffic requests, eliminating improper requests, such as inappropriate content for work. They can also act as a cache server, cutting down on outside network requests for the same object. Finally, proxy servers protect the identity of internal IP addresses, although this function can also be accomplished through a router or firewall using Network Address Translation (NAT).

Extranet

An *extranet* is an extension of a selected portion of a company's intranet to external partners. This allows a business to share information with customers, suppliers, partners, and other trusted groups while using a common set of Internet protocols to facilitate operations. Extranets can use public networks to extend their reach beyond a company's own internal network, and some form of security, typically VPN, is used to secure this channel. The use of the term *extranet* implies both privacy and security. Privacy is required for many communications, and security is needed to prevent unauthorized use and events from occurring. Both of these functions can be achieved through the use of technologies. Proper firewall management, remote access, encryption, authentication, and secure tunnels across public networks are all methods used to ensure privacy and security for extranets.

Telephony
Data and voice communications have coexisted in enterprises for decades. Recent connections inside the enterprise of Voice over IP and traditional PBX solutions increase both functionality and security risks. Specific firewalls to protect against unauthorized traffic over telephony connections are available to counter the increased risk.

VLANs
A local area network (LAN) is a set of devices with similar functionality and similar communication needs, typically co-located and operated off a single switch. This is the lowest level of a network hierarchy and defines the domain for certain protocols at the data link layer for communication. Virtual LANs use a single switch and divide it into multiple broadcast domains and/or multiple network segments, known as *trunking*. This very powerful technique allows significant network flexibility, scalability, and performance.

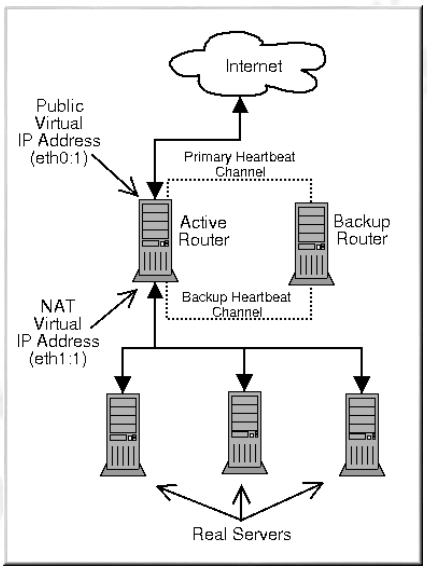
Trunking
Trunking is the process of spanning a single VLAN across multiple switches. A trunk-based connection between switches allows packets from a single VLAN to travel between switches. VLAN 10 is implemented with one trunk and VLAN 20 is implemented by the other. Hosts on different VLANs cannot communicate using trunks and are switched across the switch network. Trunks enable network administrators to set up VLANs across multiple switches with minimal effort. With a combination of trunks and VLANs, network administrators can subnet a network by user functionality without regard to host location on the network or the need to recable machines.

Security Implications
VLANs are used to divide a single network into multiple subnets based on functionality. This permit engineering and accounting, for example, to share a switch because of proximity and yet have separate traffic domains. The physical placement of equipment and cables is logically and programmatically separated so adjacent ports on a switch can reference separate subnets. This prevents unauthorized use of physically close devices through separate subnets, but the same equipment. VLANs also allow a network administrator to define a VLAN that has no users and map all of the unused ports to this

VLAN. Then if an unauthorized user should gain access to the equipment, he will be unable to use unused ports, as those ports will be securely defined to nothing. Both a purpose and a security strength of VLANs is that systems on separate VLANs cannot directly communicate with each other.

NAT

Network Address Translation (NAT) uses two sets of IP addresses for resources—one for internal use and another for external (Internet) use. NAT was developed as a solution to the rapid depletion of IP addresses in the IPv4 address space; it has since become an Internet standard (see RFC 1631 for details). NAT is used to translate between the two addressing schemes and is typically performed at a firewall or router. This permits enterprises to use the non-routable private IP address space internally and reduces the number of external IP addresses used across the Internet. Three sets of IP addresses are defined as non-routable, which means that addresses will not be routed across the Internet. These addresses are routable internally and routers can be set to route them, but the routers across the Internet are set to discard packets sent to these addresses. This approach enables a separation of internal and external traffic and allows these addresses to be reused by anyone and everyone who wishes to do so. The three address spaces are:



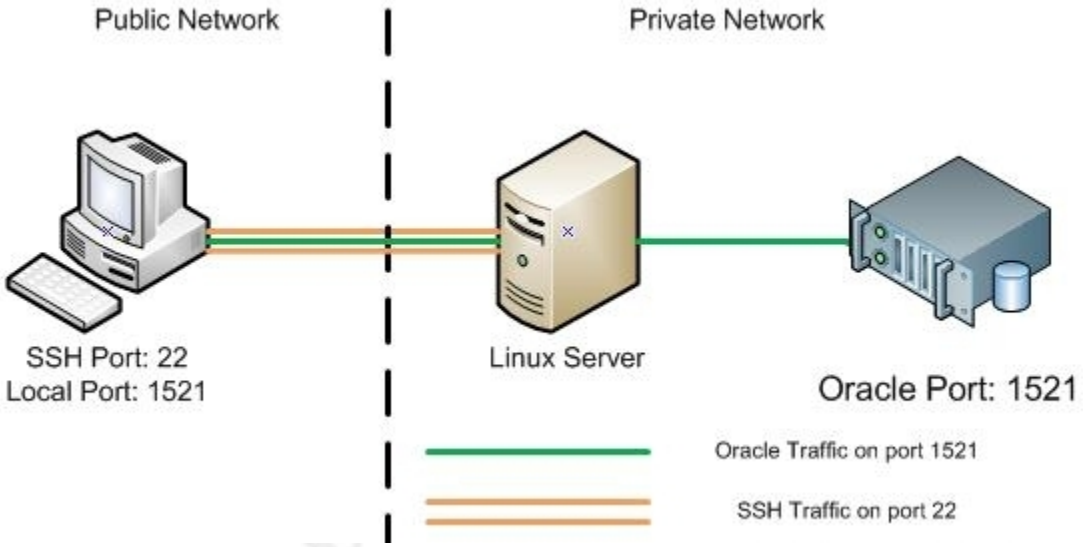
- **Class A** 10.0.0.0 – 10.255.255.255
- **Class B** 172.16.0.0 – 172.31.255.255
- **Class C** 192.168.0.0 – 192.168.255.255

The use of these addresses inside a network is unrestricted, and they function like any other IP addresses. When outside—that is, Internet-provided—resources are needed for one of these addresses, NAT is required to produce a valid external IP address for the resource. NAT operates by translating the address when traffic passes the NAT device, such as a firewall. The external addresses used are not externally mappable 1:1 to the internal addresses, for this would defeat the purpose of reuse and address-space conservation. Typically, a pool of external IP addresses is used by the NAT device, with the device keeping track of which internal address is using which external address at any given time. This provides a significant layer of security, as it makes it difficult to map the internal network structure behind a firewall and directly address it from the outside. NAT is one of the methods used for enforcing perimeter security by forcing users to access resources through defined pathways such as firewalls and gateway servers.

Tunneling

Tunneling is a method of packaging packets so that they can traverse a network in a secure, confidential manner. Tunneling involves encapsulating packets within packets, enabling dissimilar protocols to coexist in a single communication stream, as in IP traffic routed over an Asynchronous Transfer Mode (ATM) network. Tunneling also can provide significant measures of security and confidentiality through encryption and encapsulation methods. The best example of this is a VPN that is established over a public network through the use of a tunnel; connecting a firm's Boston office to its New York City (NYC) office.

Because of ease of use, low-cost hardware, and strong security, tunnels and the Internet are a combination that will see more use in the future. IPsec, VPN, and tunnels will become a major set of tools for users requiring secure network connections across public segments of networks.



Security in Transmissions

Intrusion Detection Systems

The foundation for a layered network security approach usually starts with a well secured system, regardless of the system's function (whether it's a user PC or a corporate e-mail server). A well-secured system uses up-to-date application and operating system patches, well-chosen passwords, the minimum number of services running, and restricted access to available services. On top of that foundation, you can add layers of protective measures such as antivirus products, firewalls, sniffers, and IDSs. Some of the more complicated and interesting types of network/data security devices are IDSs, which are to the network world what burglar alarms are to the physical world. The main purpose of an IDS is to identify suspicious or malicious activity, note activity that deviates from normal behavior, catalog and classify the activity, and, if possible, respond to the activity. This

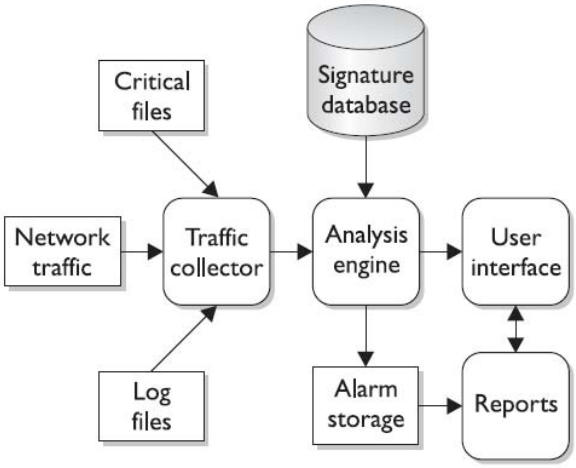
chapter looks at the history of IDSs and various types of IDSs, considers how they work and the benefits and weaknesses of specific types, and what the future might hold for these systems. You'll also look at some topics complementary to IDSs: malware protection, traffic shaping/filtering, and honey pots

IDS Overview

As mentioned, an IDS is somewhat like a burglar alarm. It watches the activity going on around it and tries to identify undesirable activity. IDSs are typically divided into two main categories, depending on how they monitor activity:

- **Host-based IDS** Examines activity on an individual system, such as a mail server, web server, or individual PC. It is concerned only with an individual system and usually has no visibility into the activity on the network or systems around it.
- **Network-based IDS** Examines activity on the network itself. It has visibility only into the traffic crossing the network link it is monitoring and typically has no idea of what is happening on individual systems.

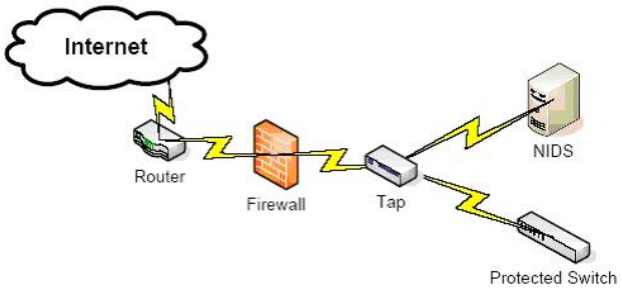
Whether or not it is network- or host-based, an IDS will typically consist of several specialized components working together, as illustrated in Figure 11-2. These components are often logical and software-based rather than physical and will vary slightly from vendor to vendor and product to product. Typically, an IDS will have the following logical components:



- **Traffic collector (or sensor):** This component collects activity/events for the IDS to examine. On host-based IDS, this could be log files, audit logs, or traffic coming to or leaving a specific system. On a network-based IDS, this is typically a mechanism for copying traffic off the network link—basically functioning as a sniffer. This component is often referred to as a sensor.
- **Analysis engine**: This component examines the collected network traffic and compares it to known patterns of suspicious or malicious activity stored in the signature database. The analysis engine is the "brains" of the IDS.
- **Signature database:** The signature database is a collection of patterns and definitions of known suspicious or malicious activity.
- **User interface and reporting:** This component interfaces with the human element, providing alerts when appropriate and giving the user a means to interact with and operate the IDS.

Most IDSs can be tuned to fit a particular environment. Certain signatures can be turned off, telling the IDS not to look for certain types of traffic.

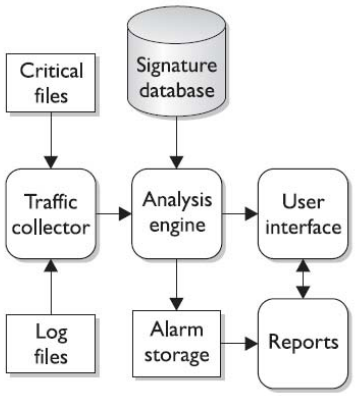
Host-based IDSs



The first IDSs were host-based and designed to examine activity only on a specific host. A host-based IDS (HIDS) examines log files, audit trails, and network traffic coming in to or leaving a specific host. HIDS can operate in *real time*, looking for activity as it occurs, or in *batch mode*, looking for activity on a periodic basis. Host-based systems are typically self-contained, but many of the newer commercial products have been designed to report to and be managed by a central system. Host-based systems also take local system resources to operate. In other words, a HIDS will use up some of the memory and CPU cycles of the system it is protecting. Early versions of HIDS ran in batch mode, looking for suspicious activity on an hourly or daily basis, and typically looked only for specific events in the system's log files. As processor speeds increased, later versions of HIDSs looked through the log files in real time and even added the ability to examine the data traffic the host was generating and receiving.

Most HIDS focus on the log files or audit trails generated by the local operating system. On UNIX systems, the examined logs usually include those created by syslog such as messages, kernel logs, and error logs. On Windows systems, the examined logs are typically the three event logs: Application, System, and Security. Some HIDS can cover specific applications, such as FTP or web services, by examining the logs produced by those specific applications or examining the traffic from the services themselves. Within the log files, the HIDS is looking for certain activities that typify hostile actions or misuse, such as the following:

- Logins at odd hours
- Login authentication failures
- Additions of new user accounts
- Modification or access of critical system files
- Modification or removal of binary files (executables)
- Starting or stopping processes
- Privilege escalation
- Use of certain programs



In general, most HIDSs will operate in a very similar fashion. (Figure 11-3 shows the logical layout of a HIDS.) By considering the function and activity of

each component, you can gain some insight into how HIDSs operate.

The *analysis engine* is perhaps the most important component of the IDS, as it must decide what activity is "okay" and what activity is "bad." The analysis engine is a sophisticated decision and pattern-matching mechanism—it looks at the information provided by the traffic collector and tries to match it against known patterns of activity stored in the signature database. If the activity matches a known pattern, the analysis engine can react, usually by issuing an alert or alarm. An analysis engine may also be capable of remembering how the activity it is looking at right now compares to traffic it has already seen or may see in the near future so that it can match more complicated, multistep malicious activity patterns. An analysis engine must also be capable of examining traffic patterns as quickly as possible, as the longer it takes to match a malicious pattern, the less time the IDS or human operator has to react to malicious traffic. Most IDS vendors build a "decision tree" into their analysis engines to expedite pattern matching.

The *signature database* is a collection of predefined activity patterns that have already been identified and categorized—patterns that typically indicate suspicious or malicious activity. When the analysis engine has a traffic pattern to examine, it will compare that pattern to the appropriate signatures in the database. The signature database can contain anywhere from a few to a few thousand signatures, depending on the vendor, type of IDS, space available on the system to store signatures, and other factors. The *user interface* is the visible component of the IDS—the part that humans interact with. The user interface varies widely depending on the product and vendor and could be anything from a detailed GUI to a simple command line. Regardless of the type and complexity, the interface is provided to allow the user to interact with the system: changing parameters, receiving alarms, tuning signatures and response patterns, and so on.

Active vs. Passive HIDS

Most IDSs can be distinguished by how they examine the activity around them and whether or not they interact with that activity. This is certainly true for HIDSs. On a *passive* system, the IDS is exactly that—it simply watches the activity, analyzes it, and generates alarms. It does not interact with the activity itself in any way, and it does not modify the defensive posture of the system to react to the traffic. A passive IDS is similar to a simple motion sensor—it generates an alarm when it matches a pattern much as the motion sensor generates an alarm when it sees movement.

An *active* IDS will contain all the same components and capabilities of the passive IDS with one critical exception—the active IDS can *react* to the activity it is analyzing. These reactions can range from something simple, such as running a script to turn a process on or off, to something as complex as modifying file permissions, terminating the offending processes, logging off specific users, and reconfiguring local capabilities to prevent specific users from logging in for the next 12 hours.

PC-based Malware Protection

In the early days of PC use, threats were limited: most home users were not connected to the Internet 24/7 through broadband connections, and the most common threat was a virus passed from computer to computer via an infected floppy disk. But things have changed dramatically over the last decade and current threats pose a much greater risk than ever before. According to SANS Internet Storm Center, the average survival time of an unpatched Windows PC on the Internet is less than 60 minutes (http://isc.sans.org/survivaltime.html). This is the estimated time before an automated probe finds the system, penetrates it, and compromises it. Automated probes from botnets and worms are not the only threats roaming the Internet—viruses and malware spread by e-mail, phishing, infected web sites that execute code on your system when you visit them, adware, spyware, and so on. Fortunately, as the threats increase in complexity and capability, so do the products designed to stop them.

Antivirus Products

Antivirus products attempt to identify, neutralize, or remove malicious programs, macros, and files. These products were initially designed to detect and remove computer viruses, though many of the antivirus products are now bundled with additional security products and features. At the present time, there is no real consensus regarding the first antivirus product. The first edition of Polish antivirus software *mks_vir* was released in 1987, and the first publicly-known neutralization of a PC virus was performed by European Bernt Fix (also known as Bernd) early in the same year. By 1990, software giants McAfee and Norton both had established commercial antivirus products.

Personal Software Firewalls

Personal firewalls are host-based protective mechanisms that monitor and control traffic passing into and out of a single system. Designed for the end user, software firewalls often have a configurable security policy that allows the user to determine what traffic is "good" and allowed to pass and what traffic is "bad" and is blocked. Software firewalls are extremely commonplace—so much so that most modern operating systems come with some type personal firewall included. For example, with the introduction of the Windows XP Professional operating system, Microsoft included a utility called the Internet Connection Firewall. Though disabled by default and hidden in the network configuration screens where most users would never find it, the Internet Connection Firewall did give users some direct control over the network traffic passing through their systems. When Service Pack 2 was launched, Microsoft renamed the Internet Connection Firewall the Windows Firewall and enabled it by default (Vista also enables the Windows firewall by default). The Windows firewall is fairly configurable; it can be set up to block all traffic, make exceptions for traffic you want to allow, and log rejected traffic for later analysis. With the introduction of the Vista operating system, Microsoft modified the Windows Firewall to make it more capable and configurable. More options were added to allow for more granular control of network traffic as well as the ability to detect when certain components are not behaving as expected. For example, if your MS Outlook client suddenly attempts to connect to a remote web server, the Windows Firewall can detect this as a deviation from normal behavior and block the unwanted traffic.

Pop-up Blocker
One of the most annoying nuisances associated with web browsing is the pop-up ad. Pop-up ads are online advertisements designed to attract web traffic to specific web sites, capture e-mail addresses, advertise a product, and perform other tasks. If you've spent more than an hour surfing the web, you've undoubtedly seen them. They're created when the web site you are visiting opens a new web browser window for the sole purpose of displaying an advertisement. Pop-up ads typically appear in front of your current browser window to catch your attention (and disrupt your browsing). Pop-up ads can range from mildly annoying, generating one or two pop-ups, to system crippling if a malicious web site attempts to open thousands of pop-up windows on your system.

Similar to the pop-up ad is the pop-under ad that opens up behind your current browser window. You won't see these ads until your current window is closed, and they are considered by some to be less annoying than pop-ups. Another form of pop-up is the hover ad that uses Dynamic HTML to appear as a floating window superimposed over your browser window. Dynamic HTML can be very CPU-intensive and can have a significant impact on the performance of older systems.

Windows Defender
As part of its ongoing efforts to help secure its PC operating systems, Microsoft created and released a free utility called Windows Defender in February 2006. The stated purpose of Windows Defender is to protect your computer from spyware and other unwanted software (http://www.microsoft.com/athome/security/spyware/software/default.mspx). Windows Defender is standard with all versions of the Vista operating system and is available via free download for Windows XP Service Pack 2 or later in both 32- and 64-bit versions. It has the following capabilities:

- **Spyware detection and removal** Windows Defender is designed to find and remove spyware and other unwanted programs that display pop-ups, modify browser or Internet settings, or steal personal information from your PC.
- **Scheduled scanning** You can schedule when you want your system to be scanned or you can run scans on demand.
- **Automatic updates** Updates to the product can be automatically downloaded and installed without user interaction.
- **Real-time protection** Processes are monitored in real time to stop spyware and malware when they first launch, attempt to install themselves, or attempt to access your PC.
- **Software Explorer** One of the more interesting capabilities within Windows Defender is the ability to examine the various programs running on your computer. Windows Defender allows you to look at programs that run automatically on startup, are currently running on your PC, or are accessing network connections on your PC. Windows Defender provides you with details such as the publisher of the software, when it was installed on your PC, whether or not the software is "good" or considered to be known malware, the file size, publication date, and other information.

- **Configurable responses** Windows Defender lets you choose what actions you want to take in response to detected threats; you can automatically disable the software, quarantine it, attempt to uninstall it, and perform other tasks.

Network-based IDSs

Network-based IDSs (NIDS) came along a few years after host-based systems. After running host-based systems for a while, many organizations grew tired of the time, energy, and expense involved with managing the first generation of these systems. The desire for a "better way" grew along with the amount of interconnectivity between systems and consequently the amount of malicious activity coming across the networks themselves.

This fueled development of a new breed of IDS designed to focus on the source for a great deal of the malicious traffic—the network itself.

The NIDS integrated very well into the concept of *perimeter security*. More and more companies began to operate their computer security like a castle or military base with attention and effort focused on securing and controlling the ways in and out—the idea being that if you could restrict and control access at the perimeter, you didn't have to worry as much about activity inside the organization. Even though the idea of a security perimeter is somewhat flawed (many security incidents originate inside the perimeter), it caught on very quickly, as it was easy to understand and devices such as firewalls, bastion hosts, and routers were available to define and secure that perimeter. The best way to secure the perimeter from outside attack is to reject all traffic from external entities, but as this is impossible and impractical to do, security personnel needed a way to let traffic in but still be able to determine whether or not the traffic was malicious. This is the problem that NIDS developers were trying to solve.

Active vs. Passive NIDSs

Most NIDSs can be distinguished by how they examine the traffic and whether or not they interact with that traffic. On a *passive* system, the IDS simply watches the traffic, analyzes it, and generates alarms. It does not interact with the traffic itself in any way, and it does not modify the defensive posture of the system to react to the traffic. A passive IDS is very similar to a simple motion sensor—it generates an alarm when it matches a pattern much as the motion sensor generates an alarm when it sees movement. An *active* IDS will contain all the same components and capabilities of the passive IDS with one critical addition—the active IDS can *react* to the traffic it is analyzing.

These reactions can range from something simple, such as sending a TCP reset message to interrupt a potential attack and disconnect a session, to something complex, such as dynamically modifying firewall rules to reject all traffic from specific source IP addresses for the next 24 hours.

Signatures

As you have probably deduced from the discussion so far, one of the critical elements of any good IDS is the signature set—the set of patterns the IDS uses to determine whether or not activity is potentially hostile. Signatures can be very simple or remarkably complicated, depending on the activity they are trying to highlight. In general, signatures can be divided into two main groups, depending on what the signature is looking for: context-based and context-based.

Content-based signatures are generally the simplest. They are designed to examine the content of such things as network packets or log entries. Content-based signatures are typically easy to build and look for simple things, such as a certain string of characters or a certain flag set in a TCP packet. Here are some example content-based signatures: • *Matching the characters* /etc/passwd *in a Telnet session.* On a UNIX system, the names of valid user accounts (and sometimes the passwords for those user accounts) are stored in a file called *passwd* located in the *etc* directory.

- *Matching a TCP packet with the synchronize, reset, and urgent flags all set within the same packet.* This combination of flags is impossible to generate under normal conditions, and the presence of all of these flags in the same packet would indicate this packet was likely created by a potential attacker for a specific purpose, such as to crash the targeted system.
- *Matching the characters* to: decode *in the header of an e-mail message.* On certain older versions of sendmail, sending an e-mail message to "decode" would cause the system to execute the contents of the e-mail.

Context-based signatures are generally more complicated, as they are designed to match large patterns of activity and examine how certain types of activity fit into the other activities going on around them. Context signatures generally address the question How does this event compare to other events that have already happened or might happen in the near future? Context-based signatures are more difficult to analyze and take more resources to match, as the IDS must be able to "remember" past events to match certain context signatures. Here are some examples of context-based signatures:

- *Match a potential intruder scanning for open web servers on a specific network.* A potential intruder may use a port scanner to look for any systems accepting connections on port 80. To match this signature, the IDS must analyze all attempted connections to port 80 and then be able to determine which connection attempts are coming from the same source but are going to multiple, different destinations.
- *Identify a Nessus scan.* Nessus is an open-source vulnerability scanner that allows security administrators (and potential attackers) to quickly examine systems for vulnerabilities. Depending on the tests chosen, Nessus will typically perform the tests in a certain order, one after the other. To be able to determine the presence of a Nessus scan, the IDS must know which tests Nessus runs as well as the typical order in which the tests are run.
- *Identify a ping flood attack.* A single ICMP packet on its own is generally regarded as harmless, certainly not worthy of an IDS signature. Yet thousands of ICMP packets

coming to a single system in a short period of time can have a devastating effect on the receiving system. By flooding a system with thousands of valid ICMP packets, an attacker can keep a target system so busy it doesn't have time to do anything else—a very effective denial-of-service attack. To identify a ping flood, the IDS must recognize each ICMP packet and keep track of how many ICMP packets different systems have received in the recent past.

False Positives and Negatives

Viewed in its simplest form, an IDS is really just looking at activity (be it host-based or network-based) and matching it against a predefined set of patterns. When it matches an activity to a specific pattern, the IDS cannot know the true intent behind that activity—whether or not it is benign or hostile—and therefore it can react only as it has been programmed to do. In most cases, this means generating an alert that must then be analyzed by a human who tries to determine the intent of the traffic from whatever information is available. When an IDS matches a pattern and generates an alarm for benign traffic, meaning the traffic was not hostile and not a threat, this is called a *false positive*. In other words, the IDS matched a pattern and raised an alarm when it didn't really need to do so. Keep in mind that the IDS can only match patterns and has no ability to determine intent behind the activity, so in some ways this is an unfair label. Technically, the IDS is functioning correctly by matching the pattern, but from a human standpoint this is not information the analyst needed to see, as it does not constitute a threat and does not require intervention.

IDS Models

In addition to being divided along the host and network lines, IDSs are often classified according to the detection model they use: anomaly or misuse. For an IDS, a model is a method for examining behavior so that the IDS can determine whether that behavior is "not normal" or in violation of established policies.

An *anomaly* detection model is the more complicated of the two. In this model, the IDS must know what "normal" behavior on the host or network being protected really is. Once the "normal" behavior baseline is established, the IDS can then go to work identifying deviations from the norm, which are further scrutinized to determine whether that activity is malicious. Building the profile of normal activity is usually done by the IDS, with some input from security administrators, and can take days to months. The IDS must be flexible and capable enough to account for things such as new systems, new users, movement of information resources, and other factors, but be sensitive enough to detect a single user illegally switching from one account to another at 3 A.M. on a Saturday.

Intrusion Prevention Systems

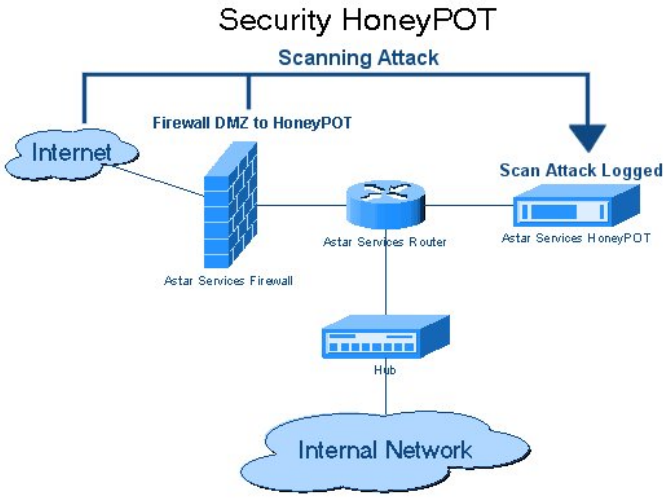
An intrusion prevention system (IPS) monitors network traffic for malicious or unwanted behavior and can block, reject, or redirect that traffic in real time. Sound familiar? It should: While many vendors will argue that an IPS is a different animal from an IDS, the truth is that most IPS are merely expansions of existing IDS capabilities. As a core

function, an IPS must be able to monitor for and detect potentially malicious network traffic, which is essentially the same function as an IDS. However, an IPS does not stop at merely monitoring traffic—it must be able to block, reject, or redirect that traffic in real time to be considered a true IPS. It must be able to stop or prevent malicious traffic from having an impact. To qualify as an IDS a system just needs to see and classify the traffic as malicious. To qualify as an IPS, the system must be able to do something about that traffic. In reality, most products that are called IDSs, including the first commercially available IDS, NetRanger, can interact with and stop malicious traffic, so the distinction between the two is often blurred. The term *intrusion prevention system* was originally coined by Andew Plato in marketing literature developed for NetworkICE, a company that was purchased by ISS and which is now part of IBM.

Honeypots and Honeynets

As is often the case, one of the best tools for information security personnel has always been knowledge. To secure and defend a network and the information systems on that network properly, security personnel need to know what they are up against. What types of attacks are being used? What tools and techniques are popular at the moment? How effective is a certain technique? What sort of impact will this tool have on my network? Often this sort of information is passed through white papers, conferences, mailing lists, or even word of mouth. In some cases, the tool developers themselves provide much of the information in the interest of promoting better security for everyone. Information is also gathered through examination and forensic analysis, often after a major incident has already occurred and information systems are already damaged.

One of the most effective techniques for collecting this type of information is to observe activity first-hand—watching an attacker as she probes, navigates, and exploits his way through a network. To accomplish this without exposing critical information systems, security researchers often use something called a *honeypot*.



A honeypot, sometimes called a *digital sandbox*, is an artificial environment where attackers can be contained and observed without putting real systems at risk. A good honeypot appears to an attacker to be a real network consisting of application servers, user systems, network traffic, and so on, but in most cases it's actually made up of one or a few systems running specialized software to simulate the user and network traffic common to most targeted networks. Figure 11-12 illustrates a simple honeypot layout in which a single system is placed on the network to deliberately attract attention from potential attackers.

There are many honeypots in use, specializing in everything from wireless to denialof-service attacks; most are run by research, government, or law enforcement organizations. Why aren't more businesses running honeypots? Quite simply, the time and cost are prohibitive. Honeypots take a lot of time and effort to manage and maintain and even more effort to sort, analyze, and classify the traffic the honeypot collects. Unless they are developing security tools, most companies focus their limited security efforts on preventing attacks, and in many cases, companies aren't even that concerned with detecting attacks as long as the attacks are blocked, are unsuccessful, and don't affect business operations. Even though honeypots can serve as a valuable resource by luring attackers away from production systems and allowing defenders to identify and thwart potential attackers before they cause any serious damage, the costs and efforts involved deter many companies from using honeypots.

Firewalls

Arguably one of the first and most important network security tools is the firewall. A *firewall* is a device that is configured to permit or deny network traffic based on an established policy or rule set. In their simplest form, firewalls are like network traffic cops; they determine which packets are allowed to pass into or out of the network perimeter. The term *firewall* was borrowed from the construction field, in which a fire wall is literally a wall meant to confine a fire or prevent a fire's spread within or between buildings. In the network security world, a firewall stops the malicious and untrusted traffic (the fire) of the Internet from spreading into your network. Firewalls control traffic flow between zones of network traffic; for example, between the Internet (a zone with no trust) and an internal network (a zone with high trust).

Proxy Servers

Though not strictly a security tool, a *proxy server* can be used to filter out undesirable traffic and prevent employees from accessing potentially hostile web sites. A proxy server takes requests from a client system and forwards it to the destination server on behalf of the client. Proxy servers can be completely transparent (these are usually called *gateways* or *tunneling proxies*), or a proxy server can modify the client request before sending it on or even serve the client's request without needing to contact the destination server. Several major categories of proxy servers are in use:

- **Anonymizing proxy** An anonymizing proxy is designed to hide information about the requesting system and make a user's web browsing experience "anonymous." This type of proxy service is often used by individuals concerned with the amount of personal information being transferred across the Internet and the use of tracking cookies and other mechanisms to track browsing activity.
- **Caching proxy** This type of proxy keeps local copies of popular client requests and is often used in large organizations to reduce bandwidth usage and increase performance. When a request is made, the proxy server first checks to see whether it has a current copy of the requested content in the cache; if it does, it services the client request immediately without having to contact the destination server. If the

content is old or the caching proxy does not have a copy of the requested content, the request is forwarded to the destination server.

- **Content filtering proxy** Content filtering proxies examine each client request and compare it to an established acceptable use policy. Requests can usually be filtered in a variety of ways including the requested URL, destination system, or domain name or by keywords in the content itself. Content filtering proxies typically support user-level authentication so access can be controlled and monitored and activity through the proxy can be logged and analyzed. This type of proxy is very popular in schools, corporate environments, and government networks.
- **Open proxy** An open proxy is essentially a proxy that is available to any Internet user and often has some anonymizing capabilities as well. This type of proxy has been the subject of some controversy with advocates for Internet privacy and freedom on one side of the argument, and law enforcement, corporations, and government entities on the other side. As open proxies are often used to circumvent corporate proxies, many corporations attempt to block the use of open proxies by their employees.
- **Reverse proxy** A reverse proxy is typically installed on the server side of a network connection, often in front of a group of web servers. The reverse proxy intercepts all incoming web requests and can perform a number of functions including traffic filtering, SSL decryption, serving of common static content such as graphics, and performing load balancing.
- **Web proxy** A web proxy is solely designed to handle web traffic and is sometimes called a *web cache*. Most web proxies are essentially specialized caching proxies.

Internet Content Filters

With the dramatic proliferation of Internet traffic and the push to provide Internet access to every desktop, many corporations have implemented content-filtering systems to protect them from employees' viewing of inappropriate or illegal content at the workplace and the subsequent complications that occur when such viewing takes place.

Internet content filtering is also popular in schools, libraries, homes, government offices, and any other environment where there is a need to limit or restrict access to undesirable content. In addition to filtering undesirable content, such as pornography, some content filters can also filter out malicious activity such as browser hijacking attempts or cross-site–scripting attacks. In many cases, content filtering is performed with or as a part of a proxy solution as the content requests can be filtered and serviced by the same device. Content can be filtered in a variety of ways, including via the requested URL, the destination system, the domain name, by keywords in the content itself, and by type of file requested.

Protocol Analyzers

A *protocol analyzer* (also known as a *packet sniffer*, *network analyzer*, or *network sniffer*) is a piece of software or an integrated software/hardware system that can capture and decode network traffic. Protocol analyzers have been popular with system administrators and security professionals for decades because they are such versatile and

useful tools for a network environment. From a security perspective, protocol analyzers can be used for a number of activities, such as the following:

- Detecting intrusions or undesirable traffic (IDS/IPS must have some type of capture and decode ability to be able to look for suspicious/malicious traffic)
- Capturing traffic during incident response or incident handling
- Looking for evidence of botnets, Trojans, and infected systems
- Looking for unusual traffic or traffic exceeding certain thresholds
- Testing encryption between systems or applications

From a network administration perspective, protocol analyzers can be used for activities such as these:

- Analyzing network problems
- Detecting misconfigured applications or misbehaving applications
- Gathering and reporting network usage and traffic statistics
- Debugging client/server communications

Regardless of the intended use, a protocol analyzer must be able to see network traffic in order to capture and decode it. A software-based protocol analyzer must be able to place the NIC it is going to use to monitor network traffic in *promiscuous mode* (sometimes called *promisc mode*). Promiscuous mode tells the NIC to process every network packet it sees regardless of the intended destination. Normally, a NIC will process only *broadcast* packets (that are going to everyone on that subnet) and packets with the NIC's Media Access Control (MAC) address as the destination address inside the packet. As a sniffer, the analyzer must process every packet crossing the wire, so the ability to place a NIC into promiscuous mode is critical.

Network Mappers

One of the biggest challenges in securing a network can be simply knowing what is connected to that network at any given point in time. For most organizations, the "network" is a constantly changing entity. While servers may remain fairly constant, user workstations, laptops, printers, and network-capable peripherals may connect to and then disconnect from the network on a daily basis, making the network at 3 AM look quite different than the network at 10 AM. To help identify devices connected to the network, many administrators use networking mapping tools.

Network mappers are tools designed to identify what devices are connected to a given network and, where possible, the operating system in use on that device. Most network mapping tools are "active" in that they generate traffic and then listen for responses to determine what devices are connected to the network. These tools typically use the ICMP or SNMP protocol for discovery and some of the more advanced tools will create a "map" of discovered devices showing their connectivity to the network in relation to other network devices. A few network mapping tools have the ability to perform device discovery passively by examining all the network traffic in an organization and noting each unique IP address and MAC address in the traffic stream.

Anti-spam
The bane of users and system administrators everywhere, *spam* is essentially unsolicited or undesired bulk electronic messages. While typically applied to e-mail, spam can be transmitted via text message to phones and mobile devices, as postings to Internet forums, and by other means. If you've ever used an e-mail account, chances are you've received spam.

Types of Attacks and Malicious Software

Attacks can be made against virtually any layer or level of software, from network protocols to applications. When an attacker finds vulnerability in a system, he exploits the weakness to attack the system. The effect of an attack depends on the attacker's intent and can result in a wide range of effects, from minor to severe. An attack on a system might not be visible on that system because the attack is actually occurring on a different system, and the data the attacker will manipulate on the second system is obtained by attacking the first system.

Avenues of Attack
A computer system is attacked for two general reasons: it is specifically targeted by the attacker, or it is a target of opportunity. In the first case, the attacker has chosen the target not because of the hardware or software the organization is running but for another reason, such as a political reason. For example, an individual in one country might attack a government system in another country to gather secret information. Or the attacker might target an organization as part of a "hacktivist" attack—the attacker could deface the web site of a company that sells fur coats because the attacker believes using animals in this way is unethical, for example. Perpetrating some sort of electronic fraud is another reason a specific system might be targeted for attack. Whatever the reason, an attack of

this nature is usually begun before the hardware and software of the organization is known.

The Steps in an Attack
Attackers are like bank robbers in the sense that they undergo an organized process when performing an attack. The steps an attacker takes in attempting to penetrate a targeted network are similar to those that a security consultant performs during a penetration test. The following outlines the common steps of the hacking process:

1. Reconnaissance (also known as profiling)
2. Scanning
3. Researching vulnerability
4. Performing the attack

Reconnaissance
The attacker can gather as much information about the organization as possible via several means, including studying the organization's own web site, looking for postings on news groups, or consulting resources such as the Securities and Exchange Commission's (SEC's) Filings & Forms (EDGAR) web site (www.sec.gov/edgar.shtml). A number of different financial reports are available through the EDGAR site that can provide information about an organization that can prove useful for an attack, especially for social engineering attacks. The attacker wants information about IP addresses, phone numbers, names of important individuals, and what networks the organization maintains. The attacker can also use tools such as Whois.Net (www.whois.net) to link IP addresses to registrants.

Scanning
The next step begins the technical part of an attack that determines what target systems are available and active. This is often done using a *ping sweep*, which simply sends a ping (an Internet Control Message Protocol echo request) to the target machine. If the machine responds, the attacker knows it is reachable. His next step is often to perform a *port scan* to help identify which ports are open, which indicates which services may be running on the target machine. The program nmap is the de facto standard for ping sweeping and port scanning. Running nmap with the **–sv** option will perform a *banner grab* in an attempt to determine the version of the software behind open ports. An alternative GUI program for Windows is SuperScan

Researching Vulnerability
After the hacker has a list of software running on the systems, he will start researching the Internet for vulnerabilities associated with that software. Numerous web sites provide information on vulnerabilities in specific application programs and operating systems. This information is valuable to administrators who need to know what problems exist and how to patch them In addition to information about specific vulnerabilities, some sites also provide tools that can be used to exploit the vulnerabilities. An attacker can search for known vulnerabilities and tools to exploit them, download the information and tools,

then use them against a site. If the administrator for the targeted system has not installed the correct patch, the attack may be successful; if the patch has been installed, the attacker will move on to the next possible vulnerability. If the administrator has installed all the appropriate patches so that all known vulnerabilities have been addressed, the attacker may have to resort to a brute-force attack, which involves calculating user ID and password combinations. Unfortunately, this type of attack, which could be easily prevented, sometimes proves successful.

Performing the Attack
Now the attacker is ready to execute an attack, which could have many different results—the system could crash, information could be stolen off the system, or a web site could be defaced. Hackers often install a backdoor and build their own user accounts with administrative privileges so that even when you do patch the system, they can still gain access.

This discussion of attack steps is by no means complete. A system can be attacked in many different ways. The driving force behind the type of attack is the attacker's objective; if activism can be accomplished by website defacement, he may consider this a sufficient attack. If the target is more sinister, such as intellectual property theft or identity theft, data theft may be the hacker's object and hence guide his attack.

Minimizing Possible Avenues of Attack
By understanding the steps an attacker can take, you can limit the exposure of your system and minimize the possible avenues an attacker can exploit. Your first step to minimize possible attacks is to ensure that all patches for the operating system and applications are installed. Many security problems, such as viruses and worms, exploit known vulnerabilities for which patches actually exist. These attacks are successful only because administrators have not taken the appropriate actions to protect their systems.

The next step is to limit the services that are running on the system. As mentioned in earlier chapters, limiting the number of services to those that are absolutely necessary provides two safeguards: it limits the possible avenues of attack (the possible services for which a vulnerability may exist and be exploited), and it reduces the number of services the administrator has to worry about patching in the first place. Another step is to limit public disclosure of private information about your organization and its computing resources. Since the attacker is after this information, don't make it easy to obtain.

Attacking Computer Systems and Networks
Although hackers and viruses receive the most attention in the news (due to the volume of these forms of attack), they are not the only methods used to attack computer systems and networks. This chapter addresses many different ways computers and networks are attacked on a daily basis. Each type of attack threatens at least one of the three security requirements: confidentiality, integrity, and availability (the CIA of security). Attacks are thus attempts by unauthorized individuals to access or modify information, to deceive the

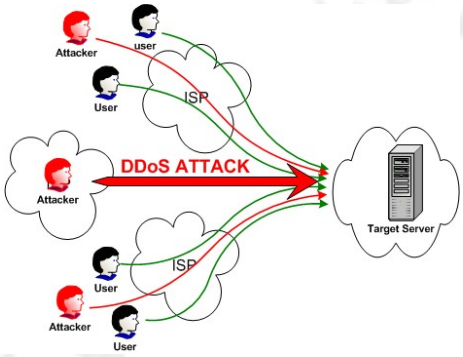
system so that an unauthorized individual can take over an authorized session, or to disrupt service to authorized users.

From a high-level standpoint, attacks on computer systems and networks can be grouped into two broad categories: attacks on specific software (such as an application or the operating system) and attacks on a specific protocol or service. Attacks on a specific application or operating system are generally possible because of an oversight in the code (and possibly in the testing of that code) or because of a flaw, or bug, in the code (again indicating a lack of thorough testing). Attacks on specific protocols or services are attempts either to take advantage of a specific feature of the protocol or service or use the protocol or service in a manner for which it was not intended. This section discusses various forms of attacks of which security professionals need to be aware.

Denial-of-Service Attacks

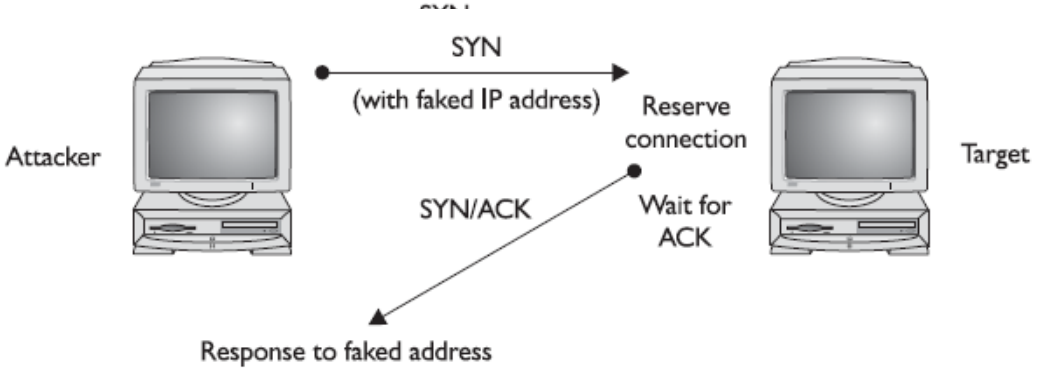
Denial-of-service (DoS) attacks can exploit a known vulnerability in a specific application or operating system, or they can attack features (or weaknesses) in specific protocols or services. In a DoS attack, the attacker attempts to deny authorized users access either to specific information or to the computer system or network itself. This can be accomplished by crashing the system—taking it offline—or by sending so many requests that the machine is overwhelmed.

The purpose of a DoS attack can be simply to prevent access to the target system, or the attack can be used in conjunction with other actions to gain unauthorized access to a computer or network. For example, a *SYN flooding* attack can be used to prevent service to a system temporarily in order to take advantage of a trusted relationship that exists between that system and another.



SYN flooding is an example of a DoS attack that takes advantage of the way TCP/IP networks were designed to function, and it can be used to illustrate the basic principles of any DoS attack. SYN flooding uses the TCP three-way handshake that establishes a connection between two systems. Under normal circumstances, the first system sends a SYN packet to the system with which it wants to communicate. The second system responds with a SYN/ACK if it is able to accept the request. When the initial system receives the SYN/ACK from the second system, it responds with an ACK packet, and communication can then proceed. This process is shown in Figure 13-1.

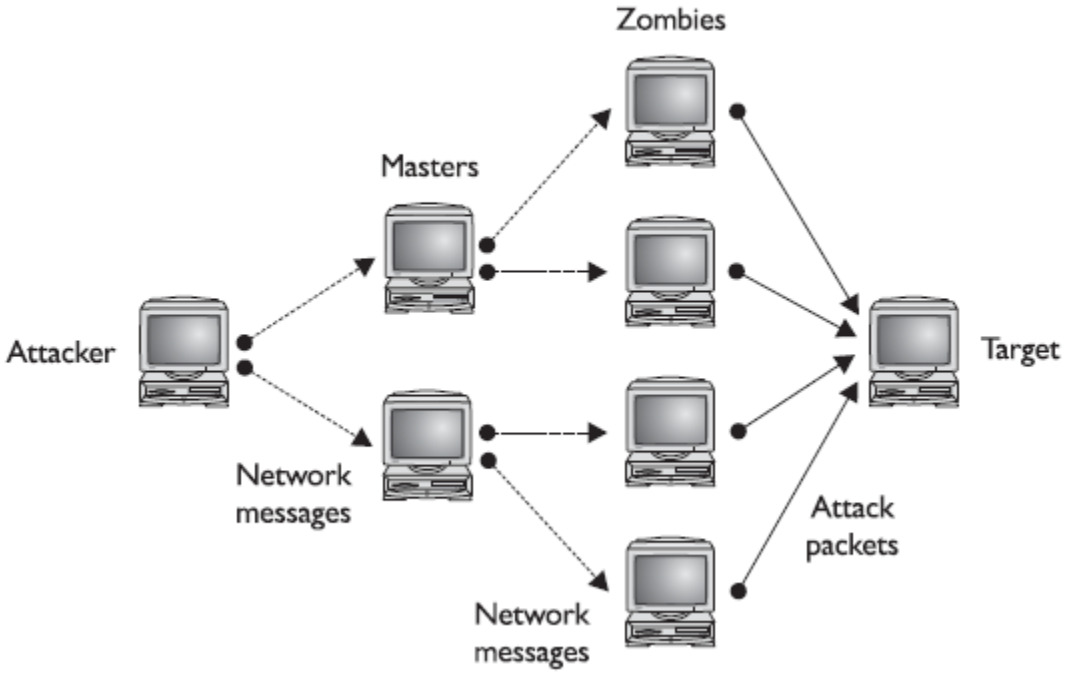
In a SYN flooding attack, the attacker sends fake communication requests to the targeted system. Each of these requests will be answered by the target system, which then waits for the third part of the handshake. Since the requests are fake (a nonexistent IP address is used in the requests, so the target system is responding to a system that doesn't exist), the



target will wait for responses that never come, as shown in Figure 13-2. The target system will drop these connections after a specific time-out period, but if the attacker sends requests faster than the time-out period eliminates them, the system will quickly be filled with requests. The number of connections a system can support is finite, so when more requests come in than can be processed, the system will soon be reserving all its connections for fake requests. At this point, any further requests are simply dropped (ignored), and legitimate users who want to connect to the target system will not be able to do so, because use of the system has been denied to them.

DoS attacks are conducted using a single attacking system. A DoS attack employing multiple attacking systems is known as a distributed denial-of-service (DDoS) attack. The goal of a DDoS attack is also to deny the use of or access to a specific service or system. DDoS attacks were made famous in 2000 with the highly publicized attacks on eBay, CNN, Amazon, and Yahoo!.
In a DDoS attack, service is denied by overwhelming the target with traffic from many different systems. A network of attack agents (sometimes called *zombies*) is created by the attacker, and upon receiving the attack command from the attacker, the attack agents commence sending a specific type of traffic against the target. If the attack network is large enough, even ordinary web traffic can quickly overwhelm the largest of sites, such as those targeted in 2000.

Creating a DDoS network is no simple task. The attack agents are not willing agents—they are systems that have been compromised and on which the DDoS attack software



has been installed. To compromise these agents, the attacker has to have gained unauthorized access to the system or tricked authorized users to run a program that installed the attack software. The creation of the attack network may in fact be a multistep process in which the attacker first compromises a few systems that are then used as *handlers* or *masters*, which in turn compromise other systems. Once the network has been created, the agents wait for an attack message that will include data on the specific target before launching the attack. One important aspect of a DDoS attack is that with just a few messages to the agents, the attacker can have a flood of messages sent against the targeted system. Figure 13-3 illustrates a DDoS network with agents and handlers.

Backdoors and Trapdoors

Backdoors were originally (and sometimes still are) nothing more than methods used by software developers to ensure that they could gain access to an application even if something were to happen in the future to prevent normal access methods. An example would be a hard-coded password that could be used to gain access to the program in the event that administrators forgot their own system password. The obvious problem with this sort of backdoor (also sometimes referred to as a *trapdoor*) is that, since it is hard-coded, it cannot be removed. Should an attacker learn of the backdoor, all systems running that software would be vulnerable to attack.

The term *backdoor* is also, and more commonly, used to refer to programs that attackers install after gaining unauthorized access to a system to ensure that they can continue to have unrestricted access to the system, even if their initial access method is discovered and blocked. Backdoors can also be installed by authorized individuals inadvertently, should they run software that contains a Trojan horse. Common backdoors include NetBus and Back Orifice. Both of these, if running on your system, can allow an attacker remote access to your system—access that allows them to perform any function on your system. A variation on the backdoor is the *rootkit*, and they are established not to gain root access but rather to ensure continued root access.

Null Sessions

Microsoft Windows systems prior to XP and Server 2003 exhibited a vulnerability in their Server Message Block system that allowed users to establish null sessions. A null session is a connection to a Windows interprocess communications share (IPC$). There is good news and bad news associated with this vulnerability. The good news is that Windows XP, Server 2003, and beyond are not susceptible to this vulnerability by default. The bad news is that the millions of previous version machines are vulnerable and patching will not solve the problem. This vulnerability can be used to glean many useful pieces of information from a machine, including user IDs, share names, registry settings, and security settings. A wide range of tools and malware use this vulnerability to achieve their aim.

To harden an affected system from the null session vulnerability requires a bit of work. The seemingly obvious path of upgrading systems to XP and beyond is not a perfect solution, for they too can be tweaked by a malicious user to become susceptible to null sessions. Although there are registry settings to restrict anonymous connections, these will not limit all types; the best method is to limit access to TCP ports 139 and 445 to only trusted users.

Sniffing

The group of protocols that make up the TCP/IP suite was designed to work in a friendly environment where everybody who connected to the network used the protocols as they were designed. The abuse of this friendly assumption is illustrated by network traffic sniffing programs, sometimes referred to as *sniffers*.

A network sniffer is a software or hardware device that is used to observe traffic as it passes through a network on shared broadcast media. The device can be used to view all traffic, or it can target a specific protocol, service, or even string of characters (looking for logins, for example). Normally, the network device that connects a computer to a network is designed to ignore all traffic that is not destined for that computer. Network sniffers ignore this friendly agreement and observe all traffic on the network, whether destined for that computer or others. A network card that is listening to all network traffic and not just its own is said to be in “promiscuous mode.” Some network sniffers are designed not just to observe all traffic but to modify traffic as well.

Network sniffers can be used by network administrators for monitoring network performance. They can be used to perform traffic analysis, for example, to determine what type of traffic is most commonly carried on the network and to determine which segments are most active. They can also be used for network bandwidth analysis and to troubleshoot certain problems (such as duplicate MAC addresses).

Spoofing

Spoofing is nothing more than making data look like it has come from a different source. This is possible in TCP/IP because of the friendly assumptions behind the protocols. When the protocols were developed, it was assumed that individuals who had access to the network layer would be privileged users who could be trusted. When a packet is sent from one system to another, it includes not only the destination IP address and port but the source IP address as well. You are supposed to fill in the source with your own address, but nothing stops you from filling in another system's address. This is one of the several forms of spoofing.

Spoofing E-Mail

In e-mail spoofing, a message is sent with a From address that differs from that of the sending system. This can be easily accomplished in several different ways using several programs. To demonstrate how simple it is to spoof an e-mail address, you can Telnet to port 25 (the port associated with e-mail) on a mail server. From there, you can fill in any address for the From and To sections of the message, whether or not the addresses are yours and whether they actually exist or not.

IP Address Spoofing

IP is designed to work so that the originators of any IP packet include their own IP address in the **From** portion of the packet. While this is the intent, nothing prevents a system from inserting a different address in the From portion of the packet. This is known as *IP address spoofing*. An IP address can be spoofed for several reasons. In a specific DoS attack known as a *smurf* attack, the attacker sends a spoofed packet to the broadcast address for a network, which distributes the packet to all systems on that network. In the smurf attack, the packet sent by the attacker to the broadcast address is an echo request with the From address forged so that it appears that another system (the target system) has made the echo request. The normal response of a system to an echo request is an echo reply, and it is used in the ping utility to let a user know whether a remote system is reachable and is responding. In the smurf attack, the request is sent to all systems on the network, so all will respond with an echo reply to the target system. The attacker has sent one packet and has been able to generate as many as 254 responses aimed at the target. Should the attacker send several of these spoofed requests, or send them to several different networks, the target can quickly become overwhelmed with the volume of echo replies it receives.

Spoofing and Sequence Numbers

How complicated the spoofing is depends heavily on several factors, including whether the traffic is encrypted and where the attacker is located relative to the target. Spoofing

attacks from inside a network, for example, are much easier to perform than attacks from outside of the network, because the inside attacker can observe the traffic to and from the target and can do a better job of formulating the necessary packets.

Formulating the packets is more complicated for external attackers because a sequence number is associated with TCP packets. A sequence number is a 32-bit number established by the host that is incremented for each packet sent. Packets are not guaranteed to be received in order, and the sequence number can be used to help reorder packets as they are received and to refer to packets that may have been lost in transmission.

Man-in-the-Middle Attacks
A man-in-the-middle attack, as the name implies, generally occurs when attackers are able to place themselves in the middle of two other hosts that are communicating. Ideally, this is done by ensuring that all communication going to or from the target host is routed through the attacker's host (which can be accomplished if the attacker can compromise the router for the target host). The attacker can then observe all traffic before relaying it and can actually modify or block traffic. To the target host, it appears that communication is occurring normally, since all expected replies are received.

The amount of information that can be obtained in a man-in-the-middle attack will obviously be limited if the communication is encrypted. Even in this case, however, sensitive information can still be obtained, since knowing what communication is being conducted, and between which individuals, may in fact provide information that is valuable in certain circumstances.

Man-in-the-Middle Attacks on Encrypted Traffic
The term "man-in-the-middle attack" is sometimes used to refer to a more specific type of attack—one in which the encrypted traffic issue is addressed. Public-key encryption, requires the use of two keys: your public key, which anybody can use to encrypt or "lock" your message, and your private key, which only you know and which is used to "unlock" or decrypt a message locked with your public key.

Replay Attacks
A *replay attack* occurs when the attacker captures a portion of a communication between two parties and retransmits it at a later time. For example, an attacker might replay a series of commands and codes used in a financial transaction to cause the transaction to be conducted multiple times. Generally replay attacks are associated with attempts to circumvent authentication mechanisms, such as the capturing and reuse of a certificate or ticket.

TCP/IP Hijacking
TCP/IP hijacking and *session hijacking* are terms used to refer to the process of taking control of an already existing session between a client and a server. The advantage to an attacker of hijacking over attempting to penetrate a computer system or network is that

the attacker doesn't have to circumvent any authentication mechanisms, since the user has already authenticated and established the session. Once the user has completed the authentication sequence, the attacker can then usurp the session and carry on as if the attacker, and not the user, had authenticated with the system. To prevent the user from noticing anything unusual, the attacker can decide to attack the user's system and perform a DoS attack on it, taking it down so that the user, and the system, will not notice the extra traffic that is taking place.

Hijack attacks generally are used against web and Telnet sessions. Sequence numbers as they apply to spoofing also apply to session hijacking, since the hijacker will need to provide the correct sequence number to continue the appropriate sessions.

Attacks on Encryption

Cryptography is the art of "secret writing," and *encryption* is the process of transforming *plaintext* into an unreadable format known as *ciphertext* using a specific technique or algorithm. Most encryption techniques use some form of key in the encryption process. The key is used in a mathematical process to scramble the original message to arrive at the unreadable ciphertext. Another key (sometimes the same one and sometimes a different one) is used to decrypt or unscramble the ciphertext to re-create the original plaintext. The length of the key often directly relates to the strength of the encryption.

Cryptanalysis is the process of attempting to break a cryptographic system—it is an attack on the specific method used to encrypt the plaintext. Cryptographic systems can be compromised in various ways.

Weak Keys

Certain encryption algorithms may have specific keys that yield poor, or easily decrypted ciphertext. Imagine an encryption algorithm that consisted solely of a single XOR function (an exclusive OR function where two bits are compared and a 1 is returned if either of the original bits, but not both, is a 1), where the key was repeatedly used to XOR with the plaintext. A key where all bits are 0's, for example, would result in ciphertext that is the same as the original plaintext. This would obviously be a weak key for this encryption algorithm. In fact, any key with long strings of 0's would yield portions of the ciphertext that were the same as the plaintext. In this simple example, many keys could be considered weak.

Encryption algorithms used in computer systems and networks are much more complicated than a simple, single XOR function, but some algorithms have still been found to have weak keys that make cryptanalysis easier.

Exhaustive Search of Key Space

Even if the specific algorithm used to encrypt a message is complicated and has not been shown to have weak keys, the key length will still play a significant role in how easy it is to attack the method of encryption. Generally speaking, the longer a key, the harder it will be to attack. Thus, a 40-bit encryption scheme will be easier to attack using a brute-

force technique (which tests all possible keys, one by one) than a 256-bit based scheme. This is easily demonstrated by imagining a scheme that employed a 2-bit key. Even if the resulting ciphertext were completely unreadable, performing a bruteforce attack until one key is found that can decrypt the ciphertext would not take long, since only four keys are possible. Every bit that is added to the length of a key doubles the number of keys that have to be tested in a brute-force attack on the encryption. It is easy to understand why a scheme utilizing a 40-bit key would be much easier to attack than a scheme that utilized a 256-bit key.

The bottom line is simple: an exhaustive search of the keyspace will decrypt the message. The strength of the encryption method is related to the sheer size of the keyspace, which with modern algorithms is large enough to provide significant time constraints when using this method to break an encrypted message. Algorithmic complexity is also an issue with respect to brute force, and you cannot immediately compare different key lengths from different algorithms and assume relative strength.

Indirect Attacks

One of the most common ways of attacking an encryption system is to find weaknesses in mechanisms surrounding the cryptography. Examples include poor random number generators, unprotected key exchanges, keys stored on hard drives without sufficient protection, and other general programmatic errors, such as buffer overflows. In attacks that target these types of weaknesses, it is not the cryptographic algorithm itself that is being attacked, but rather the implementation of that algorithm in the real world.

Address System Attacks

Addresses control many aspects of a computer system. IP addresses can be manipulated, as shown previously, and the other address schemes can be manipulated as well. In the summer of 2008, much was made of a serious domain name system (DNS) vulnerability that required the simultaneous patching of systems by over 80 vendors. This coordinated effort closed a technical loophole in the domain name resolution infrastructure that allowed hijacking and man-in-the-middle attacks on the DNS system worldwide.

Password Guessing

The most common form of authentication is the user ID and password combination. While it is not inherently a poor mechanism for authentication, the combination can be attacked in several ways. All too often, these attacks yield favorable results for the attacker not as a result of a weakness in the scheme but usually due to the user not following good password procedures.

Poor Password Choices

The least technical of the various password-attack techniques consists of the attacker simply attempting to guess the password of an authorized user of the system or network. It is surprising how often this simple method works, and the reason it does is because people are notorious for picking poor passwords. Users need to select a password that

they can remember, so they create simple passwords, such as their birthday, their mother's maiden name, the name of their spouse or one of their children, or even simply their user ID itself. All it takes is for the attacker to obtain a valid user ID (often a simple matter, because organizations tend to use an individual's names in some combination—first letter of their first name combined with their last name, for example) and a little bit of information about the user before guessing can begin. Organizations sometimes make it even easier for attackers to obtain this sort of information by posting the names of their "management team" and other individuals, sometimes with short biographies, on their web sites.

Even if the person doesn't use some personal detail as her password, she may still get lucky, since many people use a common word for their password. Attackers can obtain lists of common passwords—a number of them exist on the Internet. Words such as "password" and "secret" have often been used as passwords. Names of favorite sports teams also often find their way onto lists of commonly used passwords.

Dictionary Attack

Another method of determining passwords is to use a password-cracking program that uses a list of dictionary of words to try to guess the password. The words can be used by themselves, or two or more smaller words can be combined to form a single possible password. A number of commercial and public-domain password-cracking programs employ a variety of methods to crack passwords, including using variations on the user ID.

The programs often permit the attacker to create various rules that tell the program how to combine words to form new possible passwords. Users commonly substitute certain numbers for specific letters. If the user wanted to use the word secret for a password, for example, the letter e could be replaced with the number 3, yielding s3cr3t. This password will not be found in the dictionary, so a pure dictionary attack would not crack it, but the password is still easy for the user to remember. If a rule were created that tried all words in the dictionary and then tried the same words substituting the number 3 for the letter e, however, the password would be cracked.

Rules can also be defined so that the cracking program will substitute special characters for other characters or combine words. The ability of the attacker to crack passwords is directly related to the method the user employs to create the password in the first place, as well as the dictionary and rules used.

Brute-Force Attack

If the user has selected a password that is not found in a dictionary, even if various numbers or special characters are substituted for letters, the only way the password can be cracked is for an attacker to attempt a brute-force attack, in which the passwordcracking program attempts all possible password combinations.

The length of the password and the size of the set of possible characters in the password will greatly affect the time a brute-force attack will take. A few years ago, this method of attack was very time consuming, since it took considerable time to generate all possible combinations. With the increase in computer speed, however, generating password combinations is much faster, making it more feasible to launch brute-force attacks against certain computer systems and networks.

A brute-force attack on a password can take place at two levels: It can attack a system where the attacker is attempting to guess the password at a login prompt, or it can attack against the list of password hashes contained in a password file. The first attack can be made more difficult if the account locks after a few failed login attempts. The second attack can be thwarted if the password file is securely maintained so that others cannot obtain a copy of it.

Hybrid Attack
A hybrid password attack is a system that combines the preceding methods. Most cracking tools have this option built in, first attempting a dictionary attack, and then moving to brute-force methods.

Birthday Attack
The birthday attack is a special type of brute-force attack that gets its name from something known as the birthday paradox, which states that in a group of at least 23 people, the chance that two individuals will have the same birthday is greater than 50 percent. Mathematically, we can use the equation 1.25k1/2 (with k equaling the size of the set of possible values), and in the birthday paradox, k would be equal to 365 (the number of possible birthdays). This same phenomenon applies to passwords, with k (number of passwords) being quite a bit larger.

Software Exploitation
An attack that takes advantage of bugs or weaknesses in software is referred to as software exploitation. These weaknesses can be the result of poor design, poor testing, or poor coding practices. They can also result from what are sometimes called “features.” An example of this might be a debugging feature, which when used during debugging might allow unauthenticated individuals to execute programs on a system. If this feature remains in the program in when the final version of the software is shipped, it creates a weakness that is just waiting to be exploited.

Buffer Overflow Attack
A common weakness that has often been exploited is a buffer overflow. A buffer overflow occurs when a program is provided more data for input than it was designed to handle. For example, what would happen if a program that asks for a 7- to 10-character phone number instead receives a string of 150 characters? Many programs will provide some error checking to ensure that this will not cause a problem. Some programs, however, cannot handle this error, and the extra characters continue to fill memory, overwriting other portions of the program. This can result in a number of problems,

including causing the program to abort or the system to crash. Under certain circumstances, the program can execute a command supplied by the attacker. Buffer overflows typically inherit the level of privilege enjoyed by the program being exploited. This is why programs that use root level access are so dangerous when exploited with a buffer overflow, as the code that will execute does so at root level access.

Malicious Code
Malicious code refers to software that has been designed for some nefarious purpose. Such software can be designed to cause damage to a system, such as by deleting all files, or it can be designed to create a backdoor in the system to grant access to unauthorized individuals. Generally the installation of malicious code is done so that it is not obvious to the authorized users. Several different types of malicious software can be used, such as viruses, Trojan horses, logic bombs, spyware, and worms, and they differ in the ways they are installed and their purposes.

Viruses
The best-known type of malicious code is the virus. Much has been written about viruses as a result of several high-profile security events that involved them. A virus is a piece of malicious code that replicates by attaching itself to another piece of executable code. When the other executable code is run, the virus also executes and has the opportunity to infect other files and perform any other nefarious actions it was designed to do. The specific way that a virus infects other files, and the type of files it infects, depends on the type of virus. The first viruses created were of two types—boot sector or program viruses.

Boot Sector Virus A boot sector virus infects the boot sector portion of either a floppy disk or a hard drive (years ago, not all computers had hard drives, and many booted from a floppy). When a computer is first turned on, a small portion of the operating system is initially loaded from hardware. This small operating system then attempts to load the rest of the operating system from a specific location (sector) on either the floppy or the hard drive. A boot sector virus infects this portion of the drive. An example of this type of virus was the Stoned virus, which moved the true Master Boot Record (MBR) from the first to the seventh sector of the first cylinder and replaced the original MBR with the virus code. When the system was turned on, the virus was first executed, which had a one-in-seven chance of displaying a message stating the computer was “stoned”; otherwise, it would not announce itself and would instead attempt to infect other boot sectors. This virus was rather tame in comparison to other viruses of its time, which were often designed to delete the entire hard drive after a period of time in which they would attempt to spread.

Program Virus A second type of virus is the program virus, which attaches itself to executable files—typically files ending in .exe or .com on Windows-based systems. The virus is attached in such a way that it is executed before the program executes. Most program viruses also hide a nefarious purpose, such as deleting the hard drive data, which is triggered by a specific event, such as a date or after a certain number of other files are

infected. Like other types of viruses, program viruses are often not detected until after they execute their malicious payload. One method that has been used to detect this sort of virus before it has an opportunity to damage a system is to calculate checksums for commonly used programs or utilities. Should the checksum for an executable ever change, it is quite likely that it is due to a virus infection.

Macro Virus In the late 1990s, another type of virus appeared that now accounts for the majority of viruses. As systems and operating systems became more powerful, the boot sector virus, which once accounted for most reported infections, became less common. Systems no longer commonly booted from floppies, which were the main method for boot sector viruses to spread. Instead, the proliferation of software that included macro-programming languages resulted in a new breed of virus—the macro virus.

Avoiding Virus Infection Always being cautious about executing programs or opening documents sent to you is a good security practice. "If you don't know where it came from or where it has been, don't open or run it" should be the basic mantra for all computer users. Another security best practice for protecting against virus infection is to install and run an antivirus program. Since these programs are designed to protect against known viruses, it is also important to maintain an up-to-date listing of virus signatures for your antivirus software. Antivirus software vendors provide this information, and administrators should stay on top of the latest updates to the list of known viruses.

Two advances in virus writing have made it more difficult for antivirus software to detect viruses. These advances are the introduction of stealth virus techniques and polymorphic viruses. A stealthy virus employs techniques to help evade being detected by antivirus software that uses checksums or other techniques. Polymorphic viruses also attempt to evade detection, but they do so by changing the virus itself (the virus "evolves"). Because the virus changes, signatures for that virus may no longer be valid, and the virus may escape detection by antivirus software.

Virus Hoaxes Viruses have caused so much damage to systems that many Internet users have become extremely cautious anytime a rumor of a new virus is heard. Many users will not connect to the Internet when they hear about a virus outbreak, just to be sure their machines don't get infected. This has given rise to virus hoaxes, in which word is spread about a new virus and the extreme danger it poses. It may warn users to not read certain files or connect to the Internet.

A good example of a virus hoax was the Good Times virus warning, which has been copied repeatedly and can still be seen in various forms today. It caused widespread panic as users read about this extremely dangerous virus, which could actually cause the processor to overheat (from being put into an "nth complexity infinite binary loop") and be destroyed. Many folks saw through this hoax, but many less experienced users did not, and they passed the warning along to all of their friends.

Trojan Horses

A Trojan horse, or simply Trojan, is a piece of software that appears to do one thing (and may, in fact, actually do that thing) but hides some other functionality. The analogy to the famous story of antiquity is very accurate. In the original case, the object appeared to be a large wooden horse, and in fact it was. At the same time, it hid something much more sinister and dangerous to the occupants of the city of Troy. As long as the horse was left outside the city walls, it could cause no damage to the inhabitants. It had to be taken in by the inhabitants, and it was inside that the hidden purpose was activated. A computer Trojan works in much the same way. Unlike a virus, which reproduces by attaching itself to other files or programs, a Trojan is a standalone program that must be copied and installed by the user—it must be "brought inside" the system by an authorized user. The challenge for the attacker is enticing the user to copy and run the program.

This generally means that the program must be disguised as something that the user would want to run—a special utility or game, for example. Once it has been copied and is inside the system, the Trojan will perform its hidden purpose with the user often still unaware of its true nature.

A good example of a Trojan is Back Orifice (BO), originally created in 1999 and now offered in several versions. BO can be attached to a number of types of programs. Once it is attached, and once an infected file is run, BO will create a way for unauthorized individuals to take over the system remotely, as if they were sitting at the console. BO is designed to work with Windows-based systems. Many Trojans communicate to the outside through a port that the Trojan opens, and this is one of the ways Trojans can be detected.

Spyware

Spyware is software that "spies" on users, recording and reporting on their activities. Typically installed without user knowledge, spyware can perform a wide range of activities. It can record keystrokes (commonly called keylogging) when the user logs onto specific web sites. It can monitor how a user applies a specific piece of software, that is, monitor attempts to cheat at games. Many spyware uses seem innocuous at first, but the unauthorized monitoring of a system can be abused very easily. In other cases, the spyware is specifically designed to steal information. Many states have passed legislation banning the unapproved installation of software, but spyware can circumvent this issue through complex and confusing end-user license agreements.

Logic Bombs

Logic bombs, unlike viruses and Trojans, are a type of malicious software that is deliberately installed, generally by an authorized user. A logic bomb is a piece of code that sits dormant for a period of time until some event invokes its malicious payload. An example of a logic bomb might be a program that is set to load and run automatically, and that periodically checks an organization's payroll or personnel database for a specific employee. If the employee is not found, the malicious payload executes, deleting vital corporate files.

If the event is a specific date or time, the program will often be referred to as a time bomb. In one famous example of a time bomb, a disgruntled employee left a time bomb in place just prior to being fired from his job. Two weeks later, thousands of client records were deleted. Police were eventually able to track the malicious code to the disgruntled ex-employee, who was prosecuted for his actions. He had hoped that the two weeks that had passed since his dismissal would have caused investigators to assume he could not have been the individual who had caused the deletion of the records.

Logic bombs are difficult to detect because they are often installed by authorized users and, in particular, have been installed by administrators who are also often responsible for security. This demonstrates the need for a separation of duties and a periodic review of all programs and services that are running on a system. It also illustrates the need to maintain an active backup program so that if your organization loses critical files to this sort of malicious code, it loses only transactions that occurred since the most recent backup and no permanent loss of data results.

Rootkits

Rootkits are a form of malware that is specifically designed to modify the operation of the operating system in some fashion to facilitate nonstandard functionality. The history of rootkits goes back to the beginning of the UNIX operating system, where they were sets of modified administrative tools. Originally designed to allow a program to take greater control over operating system function when it fails or becomes unresponsive, the technique has evolved and is used in a variety of ways. One high-profile case occurred at Sony BMG Corporation, when rootkit technology was used to provide copy protection technology on some of the company's CDs. Two major issues led to this being a complete debacle for Sony: first, the software modified systems without the user's approval; and second, the software opened a security hole on Windows-based systems, creating an exploitable vulnerability at the rootkit level. This led the Sony case to be labeled as malware, which is the most common use of rootkits.

A rootkit can do many things—in fact; it can do virtually anything that the operating system does. Rootkits modify the operating system kernel and supporting functions, changing the nature of the system's operation. Rootkits are designed to avoid, either by subversion or evasion, the security functions of the operating system to avoid detection. Rootkits act as a form of malware that can change thread priorities to boost an application's performance, perform keylogging, act as a sniffer, hide other files from other applications, or create backdoors in the authentication system. The use of rootkit functionality to hide other processes and files enables an attacker to use a portion of a computer without the user or other applications knowing what is happening. This hides exploit code from antivirus and antispyware programs, acting as a cloak of invisibility.

Worms

It was once easy to distinguish between a worm and a virus. Recently, with the introduction of new breeds of sophisticated malicious code, the distinction has blurred.

Worms are pieces of code that attempt to penetrate networks and computer systems. Once a penetration occurs, the worm will create a new copy of itself on the penetrated system.

Reproduction of a worm thus does not rely on the attachment of the virus to another piece of code or to a file, which is the definition of a virus. Viruses were generally thought of as a system-based problem, and worms were network- based. If the malicious code is sent throughout a network, it may subsequently be called a worm. The important distinction, however, is whether the code has to attach itself to something else (a virus) or if it can "survive" on its own (a worm).

Some recent examples of worms that have had high profiles include the Sobig worm of 2003, the SQL Slammer worm of 2003, the 2001 attacks of Code Red and Nimba, and the 2005 Zotob worm that took down CNN Live. Nimba was particularly impressive in that it used five different methods to spread; via e-mail, via open network shares, from browsing infected web sites, using directory traversal vulnerability of Microsoft IIS 4.0/5.0, and most impressively through the use of backdoors left by Code Red II and sadmind worms.

Application-Level Attacks

Attacks against a system can occur at the network level, at the operating system level, at the application level, or at the user level (social engineering). Early attack patterns were against the network, but most of today's attacks are aimed at the applications. This is primarily because this is where the objective of most attacks resides; in the infamous words of bank robber Willie Sutton, "because that's where the money is." In fact, many of today's attacks on systems are combinations of using vulnerabilities in networks, operating systems, and applications, all means to an end to obtain the desired objective of an attack, which is usually some form of data.

War-Dialing and War-Driving

War-dialing is the term used to describe an attacker's attempt to discover unprotected modem connections to computer systems and networks. The term's origin is the 1983 movie War Games, in which the star has his machine systematically call a sequence of phone numbers in an attempt to find a computer connected to a modem. In the case of the movie, the intent was to find a machine with games the attacker could play, though obviously an attacker could have other purposes once access is obtained. War-dialing is surprisingly successful, mostly because of rogue modems—unauthorized modems attached to computers on a network by authorized users. Generally the reason for attaching the modem is not malicious—an individual may simply want to be able to go home and then connect to the organization's network to continue working.

The problem, however, is that if a user can connect, so can an attacker. If the authorized user has not implemented any security protection, this means of access could be totally open. This is often the case. Most organizations enact strict policies against connecting unauthorized modems, but it is difficult to enforce this kind of policy. Recently, new technology has been developed to address this common backdoor into corporate

networks. Telephone firewalls have been created, which block any unauthorized modem connections into an organization. These devices make it impossible for an unauthorized modem connection to be established and can also enforce strict access policies on any authorized modems.

The term war-driving has been used to refer to the activity in which attackers wander throughout an area (often in a car) with a computer with wireless capability, searching for wireless networks they can access. Some security measures can limit an attacker's ability to succeed at this activity, but, just as in war-dialing, the individuals who set up the wireless networks don't always activate these security mechanisms.

Social Engineering

Social engineering relies on lies and misrepresentation, which an attacker uses to trick an authorized user into providing information or access the attacker would not normally be entitled to. The attacker might, for example, contact a system administrator pretending to be an authorized user, asking to have a password reset. Another common ploy is to pose as a representative from a vendor needing temporary access to perform some emergency maintenance. Social engineering also applies to physical access. Simple techniques include impersonating pizza or flower delivery personnel to gain physical access to a facility.

Attackers know that, due to poor security practices, if they can gain physical access to an office, the chances are good that, given a little unsupervised time, a user ID and password pair might be found on a notepad or sticky note. Unsupervised access might not even be required, depending on the quality of the security practices of the organization.

Auditing

Auditing, in the financial community, is done to verify the accuracy and integrity of financial records. Many standards have been established in the financial community about how to record and report a company's financial status correctly. In the computer security world, auditing serves a similar function. It is a process of assessing the security state of an organization compared against an established standard.

The important elements here are the standards. Organizations from different communities may have widely different standards, and any audit will need to consider the appropriate elements for the specific community. Audits differ from security or vulnerability assessments in that assessments measure the security posture of the organization but may do so without any mandated standards against which to compare them. In a security assessment, general security "best practices" can be used, but they may lack the regulatory teeth that standards often provide. Penetration tests can also be encountered—these tests are conducted against an organization to determine whether any holes in the organization's security can be found. The goal of the penetration test is to penetrate the security rather than measuring it against some standard. Penetration tests are often viewed as white-hat hacking in that the methods used often mirror those that attackers (often called black hats) might use.

Web Components

The usefulness of the WWW is due not just to browsers, but also to web components that enable services for end users through their browser interfaces. These components use a wide range of protocols and services to deliver the desired content to end users. From a security perspective, they offer users an easy-to-use, secure method of conducting data transfers over the Internet. Many protocols have been developed to deliver this content, although for most users, the browser handles the details. From a systems point of view, many security concerns have arisen, but they can be grouped into three main tasks:

- Securing a server that delivers content to users over the web
- Securing the transport of information between users and servers over the web
- Securing the user's computer from attack over a web connection

Protocols

When two people communicate, several things must happen for the communication to be effective: They must use a language that both parties understand, and they must correctly use the language—that is, structure and syntax—to express their thoughts. The mode of communication is a separate entity entirely, for the previous statements are important in both spoken and written forms of communication. The same requirements are present with respect to computer communications and they are addressed through protocols. Protocols refer to agreed upon sets of rules that allow different vendors to produce hardware and software that can interoperate with hardware and software developed by other vendors. Because of the worldwide nature of the Internet, protocols are very important and form the basis by which all the separate parts can work together. The specific instantiation of protocols is done through hardware and software components. The majority of this chapter will concentrate on protocols related to the Internet as instantiated by software components.

Encryption (SSL and TLS)

Secure Sockets Layer (SSL) is a general-purpose protocol developed by Netscape for managing the encryption of information being transmitted over the Internet. It began as a competitive feature to drive sales of Netscape's web server product, which could then send information securely to end users. This early vision of securing the transmission channel between the web server and the browser became an Internet standard.

Today, SSL is almost ubiquitous with respect to e-commerce—all browsers support it as do web servers, and virtually all sensitive financial traffic from e-commerce web sites uses this method to protect information in transit between web servers and browsers. The Internet Engineering Task Force (IETF) embraced SSL in 1996 through a series of RFCs and named the group Transport Layer Security (TLS). Starting with SSL 3.0, in 1999 the IETF issued RFC 2246, "TLS Protocol Version 1.0," followed by RFC 2712, which added Kerberos authentication, and then RFCs 2817 and 2818, which extended TLS to HTTP version 1.1 (HTTP/1.1). Although SSL has been through several versions,

TLS begins with an equivalency to SSL 3.0, so today SSL and TLS are essentially the same although not interchangeable. SSL/TLS is a series of functions that exist in the OSI (Open System Interconnection) model between the application layer and the transport and network layers. The goal of TCP is to send an unauthenticated error-free stream of information between two computers. SSL/TLS adds message integrity and authentication functionality to TCP through the use of cryptographic methods. Because cryptographic methods are an ever-evolving field, and because both parties must agree on an implementation method, SSL/TLS has embraced an open, extensible, and adaptable method to allow flexibility and strength.

When two programs initiate an SSL/TLS connection, one of their first tasks is to compare available protocols and agree on an appropriate common cryptographic protocol for use in this particular communication. As SSL/TLS can use separate algorithms and methods for encryption, authentication, and data integrity, each of these is negotiated and determined depending upon need at the beginning of a communication.

How SSL/TLS Works

SSL/TLS uses a wide range of cryptographic protocols. To use these protocols effectively between a client and a server, an agreement must be reached on which protocol to use via the SSL handshake process. The process begins with a client request for a secure connection and a server's response. The questions asked and answered are which protocol and which cryptographic algorithm will be used. For the client and server to communicate, both sides must agree on a commonly held protocol (SSL v1, v2, v3, or TLS v1). Commonly available cryptographic algorithms include Diffie-Hellman and RSA. The next step is to exchange certificates and keys as necessary to enable authentication. Authentication was a one-way process for SSL v1 and v2 with only the server providing authentication. In SSL v3/TLS, mutual authentication of both client and server is possible.

The certificate exchange is via X.509 certificates, and public key cryptography is used to establish authentication. Once authentication is established, the channel is secured with symmetric key cryptographic methods and hashes, typically RC4 or 3DES for symmetric key and MD5 or SHA-1 for the hash functions.

The Web (HTTP and HTTPS)

HTTP is used for the transfer of hyperlinked data over the Internet, from web servers to browsers. When a user types a URL such as http://www.example.com into a browser, the http:// portion indicates that the desired method of data transfer is HTTP. Although it was initially created just for HTML pages, today many protocols deliver content over this connection protocol. HTTP traffic takes place over TCP port 80 by default, and this port is typically left open on firewalls because of the extensive use of HTTP.

One of the primary drivers behind the development of SSL/TLS was the desire to hide the complexities of cryptography from end users. When using an SSL/TLS-enabled

browser, this can be done simply by requesting a secure connection from a web server instead of nonsecure connection. With respect to HTTP connections, this is as simple as using https:// in place of http://.

When a browser is SSL/TLS-aware, the entry of an SSL/TLS-based protocol will cause the browser to perform the necessary negotiations with the web server to establish the required level of security. Once these negotiations have been completed and the session is secured by a session key, a closed padlock icon is displayed in the lower right of the screen to indicate that the session is secure. If the protocol is https:, your connection is secure; if it is http:, then the connection is carried by plaintext for anyone to see. As the tiny padlock placed in the lower-right corner of the screen could have been missed, Microsoft moved it to an obvious position next to the URL in Internet Explorer 7. Another new security feature that begins with Internet Explorer 7 and Firefox 3 is the use of high assurance SSL, a combination of an extended validation SSL certificate and a high security browser. If a high security browser, Internet Explorer 7 or Firefox 3 and beyond, establish a connection with a vendor that has registered with a certificate authority for an extended validation SSL certificate, then the URL box will be colored green and the box next to it will display the registered entity and additional validation information when clicked. These improvements were in response to phishing sites and online fraud, and although they require additional costs and registration on the part of the vendors, this is a modest up-front cost to help reduce fraud and provide confidence to customers.

One important note on SSL certificate-based security is the concept of single- versus dual-sided authentication. The vast majority of SSL connections are single-sided, meaning that only the identity of the server side is vouched for via a certificate. The client is typically not identified by certificate, mainly because of the number of clients and corresponding PKI issues. A single-sided SSL secured conversation can be attacked using a man-in-the-middle attack by capturing all the traffic and relaying responses. Dual-sided SSL would prevent this attack mechanism, yet the management of every client needing to obtain and maintain a certificate makes this practically infeasible with the current PKI available to most end users.

The objective of enabling cryptographic methods in this fashion is to make it easy for end users to use these protocols. SSL/TLS is designed to be protocol agnostic. Although designed to run on top of TCP/IP, it can operate on top of other lower level protocols, such as X.25. SSL/TLS requires a reliable lower level protocol, so it is not designed and cannot properly function on top of a non-reliable protocol such as the User Datagram Protocol (UDP). Even with this limitation, SSL/TLS has been used to secure many common TCP/IP-based services

Directory Services (DAP and LDAP)
A directory is a data storage mechanism similar to a database, but it has several distinct differences designed to provide efficient data retrieval services compared to standard database mechanisms. A directory is designed and optimized for reading data, offering

very fast search and retrieval operations. The types of information stored in a directory tend to be descriptive attribute data. A directory offers a static view of data that can be changed without a complex update transaction. The data is hierarchically described in a treelike structure, and a network interface for reading is typical. Common uses of directories include e-mail address lists, domain server data, and resource maps of network resources.

To enable interoperability, the X.500 standard was created as a standard for directory services. The primary method for accessing an X.500 directory is through the Directory Access Protocol (DAP), a heavyweight protocol that is difficult to implement completely, especially on PCs and more constrained platforms. This led to the Lightweight Directory Access Protocol (LDAP), which contains the most commonly used functionality. LDAP can interface with X.500 services, and, most importantly, LDAP can be used over TCP with significantly less computing resources than a full X.500 implementation.

LDAP offers all of the functionality most directories need and is easier and more economical to implement, hence LDAP has become the Internet standard for directory services. LDAP standards are governed by two separate entities depending upon use: The International Telecommunication Union (ITU) governs the X.500 standard, and LDAP is governed for Internet use by the IETF. Many RFCs apply to LDAP functionality, but some of the most important are RFCs 2251 through 2256 and RFCs 2829 and 2830.

SSL/TLS LDAP

LDAP over TCP is a plaintext protocol, meaning data is passed in the clear and is susceptible to eavesdropping. Encryption can be used to remedy this problem, and the application of SSL/TLS-based service will protect directory queries and replies from eavesdroppers. SSL/TLS provides several important functions to LDAP services. It can establish the identity of a data source through the use of certificates, and it can also provide for the integrity and confidentiality of the data being presented from an LDAP source. As LDAP and SSL/TLS are two separate independent protocols, interoperability is more a function of correct setup than anything else. To achieve LDAP over SSL/TLS, the typical setup is to establish an SSL/TLS connection and then open an LDAP connection over the protected channel. To do this requires that both the client and the server be enabled for SSL/TLS. In the case of the client, most browsers are already enabled. In the case of an LDAP server, this specific function must be enabled by a system administrator. As this setup initially is complicated, it’s definitely a task for a competent system administrator.

Once an LDAP server is set up to function over an SSL/TLS connection, it operates as it always has. The LDAP server responds to specific queries with the data returned from a node in the search. The SSL/TLS functionality operates to secure the channel of communication, and it is transparent to the data flow from the user’s perspective. From the outside, SSL/TLS prevents observation of the data request and response, ensuring confidentiality.

File Transfer (FTP and SFTP)
One of the original intended uses of the Internet was to transfer files from one machine to another in a simple, secure, and reliable fashion, which was needed by scientific researchers. Today, file transfers represent downloads of music content, reports, and other data sets from other computer systems to a PC-based client. Until 1995, the majority of Internet traffic was file transfers. With all of this need, a protocol was necessary so that two computers could agree on how to send and receive data. As such, FTP is one of the older protocols.

FTP
FTP is an application-level protocol that operates over a wide range of lower level protocols. FTP is embedded in most operating systems and provides a method of transferring files from a sender to a receiver. Most FTP implementations are designed to operate both ways, sending and receiving, and can enable remote file operations over a TCP/IP connection. FTP clients are used to initiate transactions and FTP servers are used to respond to transaction requests. The actual request can be either to upload (send data from client to server) or download (send data from server to client).

Clients for FTP on a PC can range from an application program to the command line ftp program in Windows/DOS to most browsers. To open an FTP data store in a browser, you can enter **ftp://url** in the browser's address field to indicate that you want to see the data associated with the URL via an FTP session—the browser handles the details. File transfers via FTP can be either binary or in text mode, but in either case, they are in plaintext across the network.

Blind FTP (Anonymous FTP)
To access resources on a computer, an account must be used to allow the operating system level authorization function to work. In the case of an FTP server, you may not wish to control who gets the information, so a standard account called anonymous exists.

This allows unlimited public access to the files and is commonly used when you want to have unlimited distribution. On a server, access permissions can be established to allow only downloading or only uploading or both, depending on the system's function. As FTP can be used to allow anyone access to upload files to a server, it is considered a security risk and is commonly implemented on specialized servers isolated from other critical functions. As FTP servers can present a security risk, they are typically not permitted on workstations and are disabled on servers without need for this functionality.

SFTP
FTP operates in a plaintext mode, so an eavesdropper can observe the data being passed. If confidential transfer is required, Secure FTP (SFTP) utilizes both the Secure Shell (SSH) protocol and FTP to accomplish this task. SFTP is an application program that encodes both the commands and the data being passed and requires SFTP to be on both the client and the server. SFTP is not interoperable with standard FTP—the encrypted commands cannot be read by the standard FTP server program. To establish SFTP data

transfers, the server must be enabled with the SFTP program, and then clients can access the server provided they have the correct credentials. One of the first SFTP operations is the same as that of FTP: an identification function that uses a username and an authorization function that uses a password. There is no anonymous SFTP account by definition, so access is established and controlled from the server using standard access control lists (ACLs), IDs, and passwords.

Vulnerabilities

Modern encryption technology can provide significant levels of privacy, up to military grade secrecy. The use of protocols such as SSL/TLS provide a convenient method for end users to use cryptography without having to understand how it works. This can result in complacency the impression that once SSL/TLS is enabled, the user is safe, but this is not necessarily the case. If a Trojan program is recording keystrokes and sending the information to another unauthorized user, for example, SSL/TLS cannot prevent the security breach. If the user is connecting to an untrustworthy site, the mere fact that the connection is secure does not prevent the other site from running a scam. Using SSL/TLS and other encryption methods will not guard against your credit card information being "lost" by a company with which you do business, as in the egghead.com credit card hack of 2000. In December 2000, egghead.com's credit card database was hacked, and as many as 3.7 million credit card numbers were exposed. Other similar stories include 55,000 credit card records being compromised by creditcards.com in 2000 and more than 300,000 records being compromised by the CD Universe hack in 1999.

The key to understanding what is protected and where it is protected requires an understanding of what these protocols can and cannot do. The SSL/TLS suite can protect data in transit, but not on either end in storage. It can authenticate users and servers, provided that the certificate mechanisms are established and used by both parties. Properly set up and used, SSL/TLS can provide a very secure method of authentication, followed by confidentiality in data transfers and data integrity checking. But again, all of this occurs during transit, and the protection ends once the data is stored.

Code-Based Vulnerabilities

The ability to connect many machines together to transfer data is what makes the Internet so functional for so many users. Browsers enable much of this functionality, and as the types of data have grown on the Internet, browser functionality has grown as well. But not all functions can be anticipated or included in each browser release, so the idea of extending browser functions through plug-ins became a standard. Browsers can perform many types of data transfer, and in some cases, additional helper programs, or plug-ins, can increase functionality for specific types of data transfers. In other cases, separate application programs may be called by a browser to handle the data being transferred. Common examples of these plug-ins and programs include Shockwave plug-ins, RealOne player (both plug-in and standalone application), Windows Media Player, and Adobe Acrobat (both plug-in and standalone). The richness that enables the desired functionality of the Internet has also spawned some additional types of interfaces in the form of ActiveX components and Java applets.

Buffer Overflows

One of the most common exploits used to hack into software is the buffer overflow. The buffer overflow is a result of poor coding practices on the part of software programmers—when any program reads input into a buffer (an area of memory) and does not validate the input for correct length, the potential for a buffer overflow exists. The buffer overflow vulnerability occurs when an application can accept more input than it has assigned storage space and the input data overwrites other program areas. The exploit concept is simple: A cracker writes an executable program that performs some action on the target machine and appends this code fragment to a legitimate response to a program on the target machine. When the target machine reads through the too long response, a buffer overflow condition causes the original program to fail. The extra malicious code fragment is now in the machine's memory, awaiting execution. If the cracker executed it correctly, the program will skip into the cracker's code, running it instead of crashing.

Java and JavaScript

Java is a computer language invented by Sun Microsystems as an alternative to Microsoft's development languages. Designed to be platform-independent and based on C, Java offered a low learning curve and a way of implementing programs across an enterprise, independent of platform. Although platform independence never fully materialized, and the pace of Java language development was slowed by Sun, Java has found itself to be a leader in object-oriented programming languages. Java, and its close cousin JavaScript, operates through an interpreter called a Java Virtual Machine (JVM) on each platform that interprets the Java code, and this JVM enables the program's functionality for the specific platform. This reliance on an interpretive step has led to performance issues, and Java is still plagued by poor performance when compared to most other languages. Security was one of the touted advantages of Java, but in reality, security is not a built-in function but an afterthought and is implemented independent of the language core. This all being said, properly coded Java can operate at reasonable rates, and when properly designed can act in a secure fashion.

These facts have led to the wide dependence on Java for much of the server-side coding for e-commerce and other web-enabled functionality. Servers can add CPUs to address speed concerns, and the low learning curve has proven cost efficient for enterprises. Java was initially designed to be used in trusted environments, and when it moved to the Internet for general use, safety became one of its much-hyped benefits. Java has many safety features, such as type checking and garbage collection, that actually improve a program's ability to run safely on a machine and not cause operating system–level failures. This isolates the user from many common forms of operating system faults that can end in the "blue screen of death" in a Windows environment, where the operating system crashes and forces a reboot of the system. Safety is not security, however, and although safe, a malicious Java program can still cause significant damage to a system.

JavaScript is a form of Java designed to be operated within a browser instance. The primary purpose of JavaScript is to enable features such as validation of forms before they are submitted to the server. Enterprising programmers found many other uses for

JavaScript, such as manipulating the browser history files, now prohibited by design. JavaScript actually runs within the browser and the code is executed by the browser itself. This has led to compatibility problems, and not just between vendors, such as Microsoft and Mozilla, but between browser versions. Security settings in Internet Explorer are done by a series of zones, allowing differing level of control over .Net functionality, ActiveX functionality, and Java functionality. Unfortunately, these settings can be changed by a Trojan program, altering the browser without alerting the user and lowering the security settings. In Firefox, using the NoScript add-in is a solution to this, but the reduced functionality leads to other issues, and requires more diligent user intervention.

ActiveX
ActiveX is the name given to a broad collection of APIs, protocols, and programs developed by Microsoft to download and execute code automatically over an Internet-based channel. The code is bundled together into an ActiveX control with an .ocx extension.

These controls are referenced in HTML using the **<object>** tag. ActiveX is a tool for the Windows environment and can be extremely powerful. It can do simple things, such as enable a browser to display a custom type of information in a particular way, and it can also perform complex tasks, such as update the operating system and application programs. This range of abilities gives ActiveX a lot of power, but this power can be abused as well as used for good purposes.

CGI
The Common Gateway Interface (CGI) was the original method for having a web server execute a program outside the web server process, yet on the same server. The intent was to pass information via environment variables to an independent program, execute the program, and return the results to the web server for display. Web servers are presentation and display engines, and they provide less than stellar results when used for other purposes. For example, a web server instance can have numerous independent connections, and a program failure that results in a process bounce can affect multiple users if it is run within the web server process. Separating any time-consuming and more risky programming cores, such as database lookups and manipulation, complex calculations, and other tasks, into separate processes was and still is a prudent idea.

Server-Side Scripts
CGI has been replaced in many web sites through newer server-side scripting technologies such as Java, Active Server Pages (ASP), ASP.Net, and PHP. All these technologies operate in much the same fashion as CGI: they allow programs to be run outside the web server and to return data to the web server to be served to end users via a web page. Each of these newer technologies has advantages and disadvantages, but all of them have stronger security models than CGI. With these security models comes reduced functionality and, as each is based on a different language, the learning curves are steeper.

Still, the need for adherence to programming fundamentals exists in these technologies code must be well designed and well written to avoid the same vulnerabilities that exist in all forms of code. Buffer overflows are still an issue. Changing languages or technologies does not eliminate the basic security problems associated with incorporating open-ended user input into code. Understanding and qualifying user responses before blindly using them programmatically is essential to the security of a system.

Cookies

Cookies are small chunks of ASCII text passed within an HTTP stream to store data temporarily in a web browser instance. Invented by Netscape, cookies pass back and forth between web server and browser and act as a mechanism to maintain state in a stateless world. State is a term that describes the dependence on previous actions. By definition, HTTP traffic served by a web server is stateless—each request is completely independent of all previous requests, and the server has no memory of previous requests. This dramatically simplifies the function of a web server, but it also significantly complicates the task of providing anything but the most basic functionality in a site. Cookies were developed to bridge this gap. Cookies are passed along with HTTP data through a Set Cookie message in the header portion of an HTTP message.

Signed Applets

Code signing was an attempt to bring the security of shrink-wrapped software to software downloaded from the Internet. Code signing works by adding a digital signature and a digital certificate to a program file to demonstrate file integrity and authenticity. The certificate identifies the author, and the digital signature contains a hash value that covers code, certificate, and signature to prove integrity, and this establishes the integrity of the code and publisher via a standard browser certificate check. The purpose of a company signing the code is to state that it considers the code it created to be safe, and it is stating that the code will not do any harm to the system (to the company's knowledge). The digital signature also tells the user that the stated company is, indeed, the creator of the code.

Browser Plug-ins

The addition of browser scripting and ActiveX components allows a browser to change how it handles data, tremendously increasing its functionality as a user interface. But all data types and all desired functionality cannot be offered through these programming technologies. Plug-ins are used to fill these gaps.

Plug-ins are small application programs that increase a browser's ability to handle new data types and add new functionality. Sometimes these plug-ins are in the form of ActiveX components, which is the form Microsoft chose for its Office plug-in, which enables a browser to manipulate various Office files, such as pivot tables from Excel, over the web. Adobe has developed Acrobat Reader, a plug-in that enables a browser to read and display Portable Document Format (PDF) files directly in a browser. PDF files

offer platform independence for printed documents and are usable across a wide array of platforms—they are a compact way to provide printed information.

Application-Based Weaknesses
Web browsers are not the only aspect of software being abused by crackers. The application software written to run on servers and serve up the content for users is also a target. Web Application Security is a fairly hot topic in security, as it has become a prime target for professional crackers. Criminal hackers typically are after some form of financial reward, whether from stolen data, stolen identity, or some form of extortion. Attacking web-based applications has proven to be a lucrative venture for several reasons. First, the target is a rich environment as company after company has developed a customer facing web presence, often including custom-coded functionality that permits customer access to back-end systems for legitimate business purposes. Second, building these custom applications to high levels of security is a difficult if not impossible feat, especially given the corporate pressure on delivery time and cost.

Open Vulnerability and Assessment Language (OVAL)
The Mitre Corporation, a government-funded research group (www.mitre.org), has done extensive research into software vulnerabilities. To enable collaboration among the many different parties involved in software development and maintenance, they have developed a taxonomy of vulnerabilities—the Common Vulnerability Enumeration (CVE). This is just one of the many related enumerations that they have developed in an effort to make machine-readable data exchanges to facilitate system management across large enterprises. The CVE led efforts such as the development of the Open Vulnerability and Assessment Language (OVAL). OVAL is comprised of two main elements, an XML-based machine readable language for describing vulnerabilities and a repository; see oval.mitre.org for more information.

In addition to the CVE and OVAL efforts, Mitre has developed a wide range of enumerations and standards designed to ease the automation of security management at the lowest levels across an enterprise. Additional efforts include

- Attack Patterns (CAPEC)
- Checklist Language (XCCDF)
- Security Content Automation (SCAP)
- Configurations (CCE)
- Platforms (CPE)
- Software Weakness Types (CWE)
- Log Format (CEE)
- Reporting (CRF)

Additional information can be obtained from the Mitre Corporation web site for Making Security Measurable at measurablesecurity.mitre.org.