NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

Network Security and Firewall

Graduation Project
COM- 400

Student: Alaa ALSHANABLEH (20000543)

Supervisor: Prof. Dr. Fahreddin MAMEDOV

Nicosia - 2004

ACKNOWLEDGMENT

"First I would like to express my thanks to my supervisor Prof. Dr.Dr. Fahreddin Mamedov. For supervising my work. Under the guidance of him I successfully overcome many difficulties and learn a lot about Network Security and Firewall. While teaching, he always helped me a lot either in my study or my life.

I would like to give special thanks to all those who supported me, and for those who taught me the true meaning of perseverance.

Special thanks for my family specially my parents, for moral and material supporting, and to help me to complete my study and to become an engineer.

Finally, I want to thank my uncle Dr. Tayseer ALshanableh and all my friends (special Ahmad Watad) who support me all the time and for spent a nice days with them in this university."

INTRODUCTION

The world of computers has changed dramatically over the past 25 years. Twenty-five years ago, most computers were centralized and managed in data centers. Computers were kept in locked rooms and links outside a site were unusual. Computer security threats were rare, and were basically concerned with insiders; these threats were well understood and dealt with using standard techniques, computers behind locked doors and accounting for all resources. Twenty-five years later, many systems are connected to the Internet. The Internet is a huge network and has no boundaries. Businesses find an increasing need to connect to the internet to take advantage of the business opportunities.

The security framework for systems with internet connections is however very different. Information on the internet can be accessed from anywhere in the world in real time. While this is good for the spread of information, it has also allowed for the proliferation of 'malicious information'. Hacker tools are now widely available on the internet. Some web sites even provides tutorials on how to hack into a system, giving details of the vulnerabilities of the different kinds of systems. It does not take an expert programmer to break into a system. Anyone with malicious intentions can search the internet for programs to break into a system which is not properly secured.

It is hence vital for businesses with connections to the internet to ensure that their networks are secure. This is important to minimize the risk of intrusions both from insiders and outsiders. Although a network cannot be 100% safe, a secure network will keep everyone but the most determined hacker out of the network. A network with a good accounting and auditing system will ensure that all activities are logged thereby enabling malicious activity to be detected.

The objective of this project is to investigate the network security and firewalls. The project consists of introduction, four chapters and conclusion.

TABLE OF CONTENTS

CHAPTER ONE
OVERVIEW OF NETWORKING

1.1 Introduction to Networking

A basic understanding of computer networks is requisite in order to understand the principles of network security. In this section, we will cover some of the foundations of computer networking. Following that, we will take a more in-depth look at TCP/IP, the network protocol suite that is used to run the Internet and many intranets.

1.2 The ISO/OSI Reference Model

The *International Standards Organization* (ISO) *Open Systems Interconnect* (OSI) Reference Model defines seven layers of communications types, and the interfaces among them. (See Figure 1.1) Each layer depends on the services provided by the layer below it, all the way down to the physical network hardware, such as the computer's network interface card, and the wires that connect the cards together.

An easy way to look at this is to compare this model with something we use daily which is the telephone. In order for you and I to talk when we are out of earshot, we need a device like a telephone. (In the ISO/OSI model, this is at the application layer.) The telephones, of course, are useless unless they have the ability to translate the sound into electronic pulses that can be transferred over wire and back again. (These functions are provided in layers below the application layer.) Finally, we get down to the physical connection, both must be plugged into an outlet that is connected to a switch that's part of the telephone system's network of switches.

If preson A places a call to person B, person A picks up the receiver, and dials person B's number. This number specifies which central office to which to send my request, and then which phone from that central office to ring. Once person B answers the phone, they begin talking, and their session has begun. Conceptually, computer networks function exactly the same way.

It isn't important to memorize the ISO/OSI Reference Model's layers; but it is useful to know that they exist, and that each layer can not work without the services provided by the layer below it.

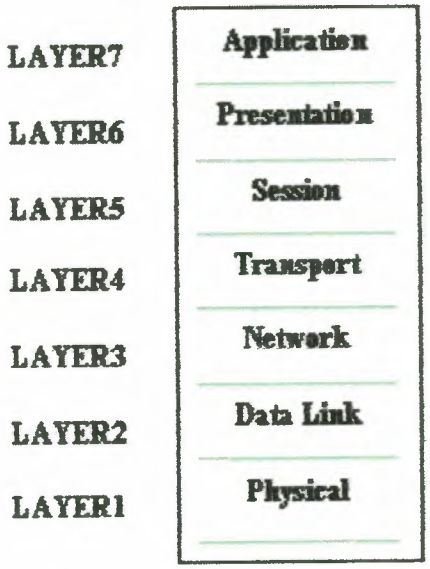

Figure 1.1. OSI Reference Model

The *physical layer* of the model consists of the actual medium through which bits are transmitted from one location to another, in other words, the fabric of the network itself. The connection between two network stations may be in the form of copper or some other electrically conductive cable, fiber optic, radio signals, microwaves, lasers, infrared, or any other medium practically suited to the environment. The OSI model makes no distinctions concerning the actual hardware involved, but the physical layer comprises every component that is needed to realize the connection. This includes any and all connectors, hubs, transceivers, network interfaces, and ancillary hardware, as well as the physical medium or cable itself, if any. This layer also includes the environmental specifications necessary to maintain the validity of the medium, as well as the method of signaling used to transmit bits to a remote location.

The *data link layer* as the interface between the network medium and the higher protocols, the data link layer is responsible for the final packaging of the upper-level binary data into discrete packets before it goes to the physical layer. Its frame is outermost on the packet and contains the basic addressing information that allows it to be transmitted to its destination. The data link layer also controls access to the network medium. This is a crucial element of local area networking because dozens of workstations may be vying for use of the same medium at any one time. Were all of these stations to transmit their packets simultaneously, the result would be chaos.

Protocols operating at this layer may also provide other services, such as error checking and correction and flow control.

The *network layer* is where the most crucial dividing line in network communications occurs, for this is the only layer that is actually concerned with the complete transmission of packets, or *protocol data units* (*PDUs*), from source to destination. The functions provided by the physical and data link layers are local. They are designed only to move the packets to the next station on the network medium. The primary task of the network layer is to provide the routing functionality by which packets can be sent across the boundaries of the local network segment to a destination that may be located on an adjacent network or on one thousands of miles away. What's more, the route actually taken by the packet must often be selected from many possible options, based on the relative efficiency of each.

The *transport layer*, as its primary function, provides the balance of the essential services not provided by the network layer protocol. A full-featured CO protocol at the network layer results in a relatively simple transport layer protocol, but as the functionality at the network layer diminishes, the complexity of the transport layer increases. The transport layer's task, therefore, is to provide whatever functions are necessary to elevate the network's *quality of service* (QOS) to a level suitable for the communications required of it.

We now arrive at the *session layer* and pass beyond all concerns for transmission reliability, error checking, flow control, and the like. All that can be done in these areas has been done by the time that the transport layer functions have been completed. The session layer is the most misunderstood service in the OSI model, and a great deal of discussion has gone into the question of whether its functions even warrant a layer of their own. Because of its name, it is often thought (mistakenly) to be concerned with the network logon procedure and related matters of security. The other common description is that it is concerned with matters of "dialogue control and dialogue separation." This is actually true, but more often than not, these expressions are left undefined in such treatments.

Sixth in line, the *presentation layer* acts as the interpreter for network communication. The presentation layer prepares the data for transmission by using one or more of a number of resources, including compression, encryption, or a complete translation of the data into a form more suitable for the currently-implemented communications methods.

Finally, the *application layer*, as the highest of the OSI levels, is tasked with providing the front-end of the computing experience for the user. The application layer is responsible for everything that the user will see, hear, and feel in the course of the networking process-everything from sending and receiving electronic mail, establishing Telnet or FTP sessions, to managing remote network resources.

1.3 Types of Networks

In this section some useful categorizations of networks are introduced:

1- Categorization by geographical coverage.

2- Categorization by topology.

1.3.1 Categorization By Geographical Coverage

Depending on the distances signals have to travel different technologies are used to run the connections. That's why it makes sense to distinguish computer networks by the area they cover.

1.3.1.1 Local Area Network (LAN)

A LAN is a network that covers a small area only: a house, a factory site, or a small number of near buildings. It has most often only one owner. However, the size restriction is by area only, and not by number! Large companies can easily have hundreds of workstations in a single LAN.

Hence all the computers are nearby, many different ways of designing the cable connection can be applied, and some methods of cabelling can be used, that would be too expensive for long distances. Local Area Networks usually have a *symmetric topology*. That's why there are many standards (namely those on symmetric topologies as star, ring, bus, etc.) that refer to LANs only.

1.3.1.2 Metropolitan Area Network

A Metropolitan Area Network (MAN) covers larger geographic areas, such as cities or school districts. By interconnecting smaller networks within a large geographic area, information is easily disseminated throughout the network. Local libraries and government agencies often use a MAN to connect to citizens and private industries.

1.3.1.3 Wide Area Network (WAN)

A WAN is a network that covers la large area; typically countries or continents. WANs are used to interconnect LANs over long distances. They usually have an *irregular topology.*

When examining a WAN the main interest is put on *transmission lines* and the *switching elements*, but not on the local "ends" of the WAN. Lines and switches together are called the communication subnet (*short: subnet*); it performs the data exchange in the network.

Besides data exchange in WANs application programs can be run. The machines that do that are referred to as hosts; Hosts perform applications in the network.

1.3.2 Categorization By Topology

1.3.2.1 Bus Topology

A *bus topology*, shown in Figure 1.2, features all networked nodes interconnected peer-to-peer using a single, open-ended cable. These ends must be terminated with a resistive load--that is, *terminating resistors*. This singe cable can support only a single channel. The cable is called the *bus*.

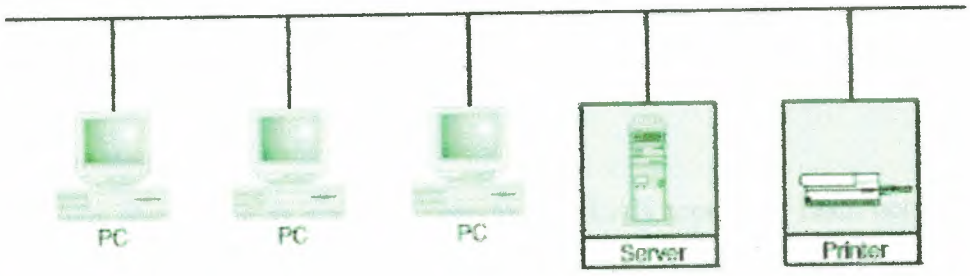

Figure 1.2. Typical bus topology.

The typical bus topology features a single cable, supported by no external electronics, that interconnects all networked nodes peer to peer. All connected devices listen to the bussed transmissions and accept those packets addressed to them. The lack of any external electronics, such as repeaters, makes bus LANs simple and inexpensive. The downside is that it also imposes severe limitations on distances, functionality, and scaleability.

1.3.2.2 Star Topology

Star topology LANs have connections to networked devices that radiate out from a common point--that is, the *hub*, as shown in Figure 1.3. Unlike ring topologies, physical or virtual, each networked device in a star topology can access the media independently. These devices have to share the hub's available bandwidth. An example of a LAN with a star topology is Ethernet.

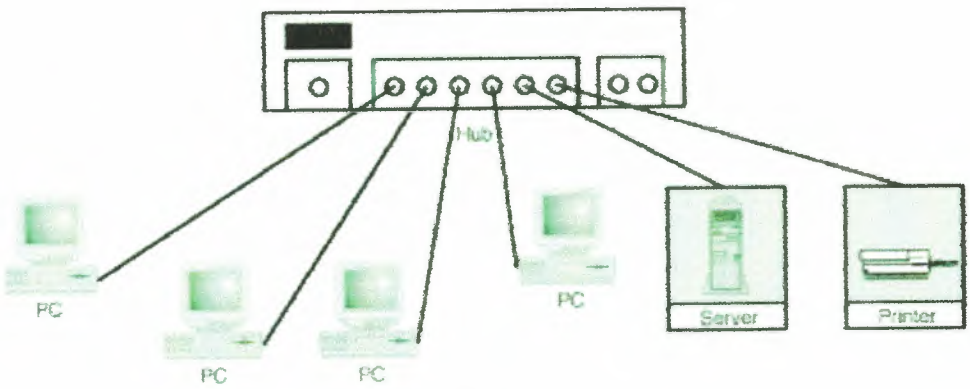

Figure 1.3. Star topology.

A small LAN with a star topology features connections that radiate out from a common point. Each connected device can initiate media access independent of the other connected devices.

1.3.2.3 Ring Topology

The *ring topology* started out as a simple peer-to-peer LAN topology. Each networked workstation had two connections: one to each of its nearest neighbors (see Figure 1.4). The interconnection had to form a physical loop, or ring. Data was transmitted unidirectionally around the ring. Each workstation acted as a repeater, accepting and responding to packets addressed to it, and forwarding on the other packets to the next workstation "downstream."

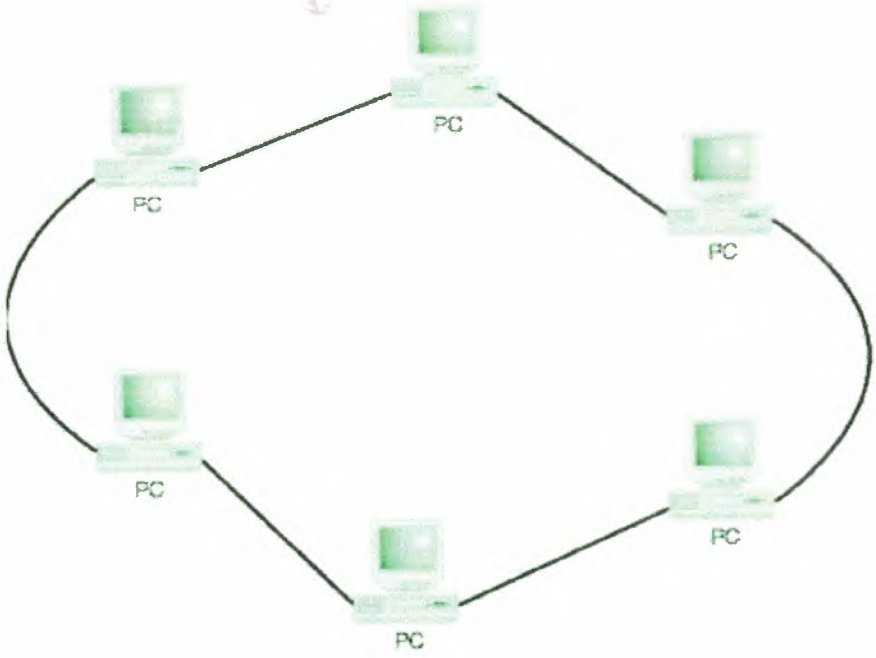

Figure 1.4. Peer-to-peer ring topology.

1.4 Network Devices

Hubs, bridges and routers are getting very intelligent, they have more and more configuration options and are increasingly complex. This is useful for additional features, but the added complexity increases the security risk. On critical subnets, it's important correctly configure network devices: only enable needed services, restrict access to configuration services by port/interface/IP address, disable broadcasts, source routing, choose strong (non default) passwords, enable logging, choose carefully who has user/enable/admin access, etc.

1.4.1 Hub

As its name implies, a *hub* is a center of activity. In more specific network terms, a hub, or concentrator, is a common wiring point for networks that are based around a star topology. Arcnet, 10base-T, and 10base-F, as well as many other proprietary network topologies, all rely on the use of hubs to connect different cable runs and to distribute data across the various segments of a network (See Figure 1.5.). Hubs basically act as a signal splitter. They take all of the signals they receive in through one port and redistribute it out through all ports. Some hubs actually regenerate weak signals before re-transmitting them. Other hubs retime the signal to provide true synchronous data communication between all ports. Hubs with multiple 10base-F connectors actually use mirrors to split the beam of light among the various ports.

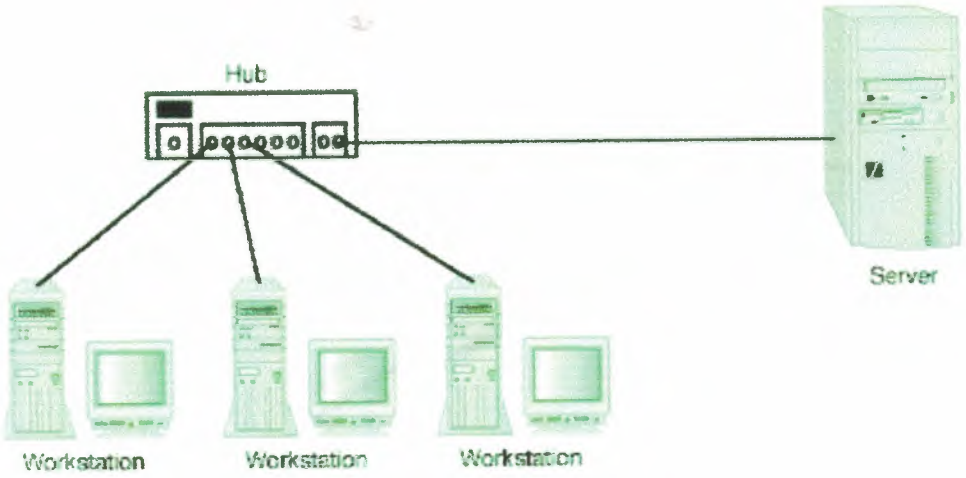

Figure 1.5. A basic diagram of a 10base-T network. Notice the hub, which is the device to which all systems initially connect.

1.4.2 Bridge

A bridge is a device that passes all data on the ethernet, token ring, or whatever type of LAN you have over the WAN to the other LAN which operate at the data link layer, connect two LANs (local area networks) together, and forward frames according to their MAC (media access control) address. Often the concept of a router is more familiar than that of a bridge; it may help to think of a bridge as a "low-level router" (routers operate at the network layer, forwarding by addresses such as an IP address).

A remote bridge connects two remote LANs (bridge 1 and 2 in Figure 1.6) over a link that is normally slow (for example, a telephone line), while a local bridge connects two locally adjacent LANs together (bridge 3 in Figure 1.6). With a local bridge, performance is an issue, but for a remote bridge, the capability to operate over a long connecting line is often more important.

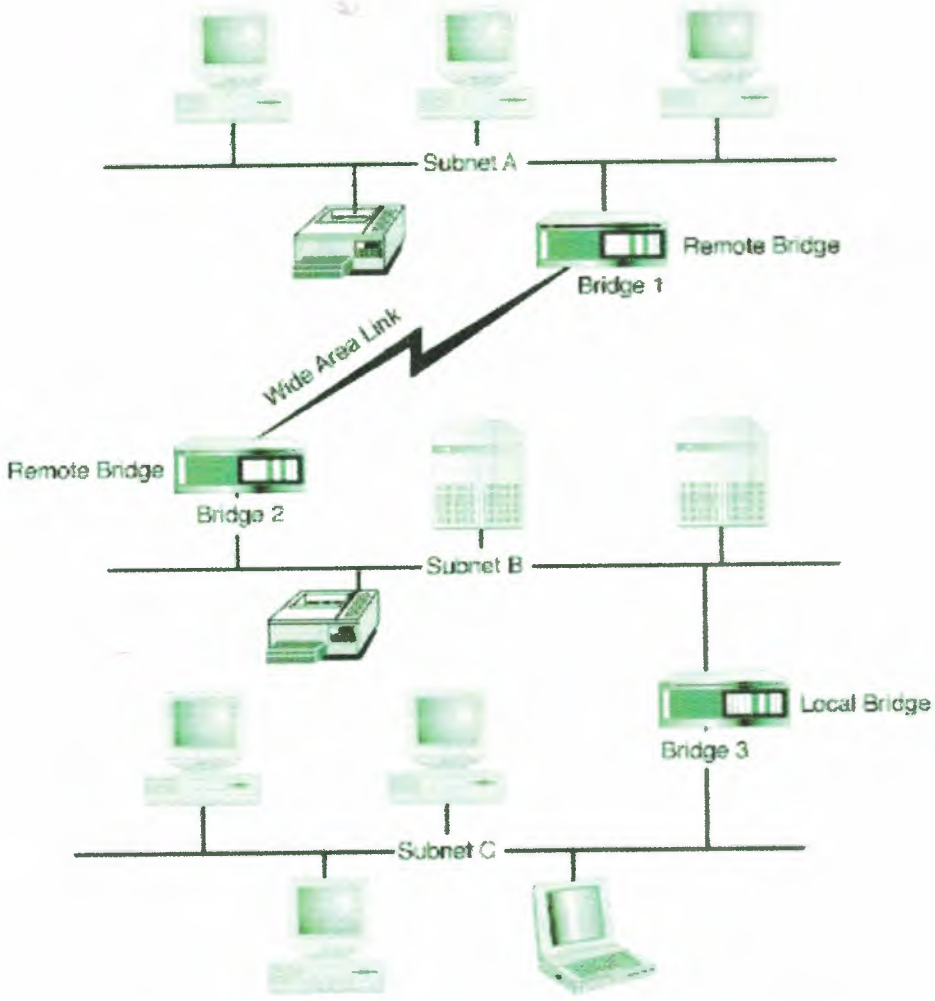

Figure 1.6. A sample network with local and remote bridges.

1.4.3 Router

Routers are devices that are installed on the LAN much as bridges are; a router connects to both the WAN and the LAN. The difference between a router and a bridge is in the way it handles the data it receives. In the bridging world, data bits on the LAN (called packets) are passed across the WAN with minimum effort on the bridge. The bridge doesn't look at the packets very closely to examine the data, because it doesn't care what the data is; it just passes the packets over to the other side of the WAN. Routers, on the other hand, examine the data sent in the packets to see whether it needs to go over the WAN or if it should stay in the LAN. Think of a data application, e-mail for instance, as if it were a letter being sent over the LAN.

1.5 What is The Internet

The Internet is the world's largest network *of networks*. When you want to access the resources offered by the Internet, you do not really connect to the Internet; you connect

to a network that is eventually connected to the Internet backbone, a network of extremely fast (and incredibly overloaded!) network components. This is an important point: the Internet is a network of *networks* -- not a network of hosts.

A simple network can be constructed using the same protocols and such that the Internet uses without actually *connecting* it to anything else. Such a basic network is shown in Figure 1.7.

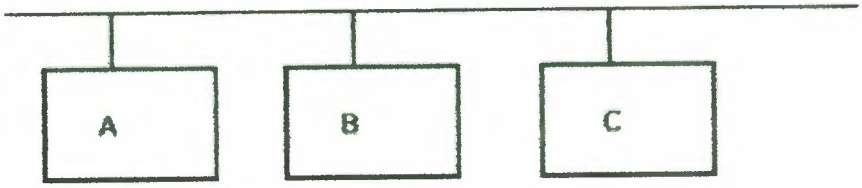

Figure 1.7. A Simple Local Area Network

Imight be allowed to put one of my hosts on one of my employer's networks. We have a number of networks, which are all connected together on a backbone , that is a network of our networks. Our backbone is then connected to other networks, one of which is to an *Internet Service Provider* (ISP) whose backbone is connected to other networks, one of which is the Internet backbone.

If you have a connection "to the Internet" through a local ISP, you are actually connecting your computer to one of their networks, which is connected to another, and so on. To use a service from my host, such as a web server, you would tell your web browser to connect to my host. Underlying services and protocols would send *packets* (small datagrams) with your query to your ISP's network, and then a network they are connected to, and so on, until it found a path to my employer's backbone, and to the exact network my host is on. My host would then respond appropriately, and the same would happen in reverse: packets would traverse all of the connections until they found their way back to your computer, and you were looking at my web page.

In Figure 1.8, the network shown in is designated "LAN 1" and shown in the bottom-right of the picture. This shows how the hosts on that network are provided connectivity to other hosts on the same LAN, within the same company, outside of the company, but in the same ISP *cloud* , and then from another ISP somewhere on the Internet.

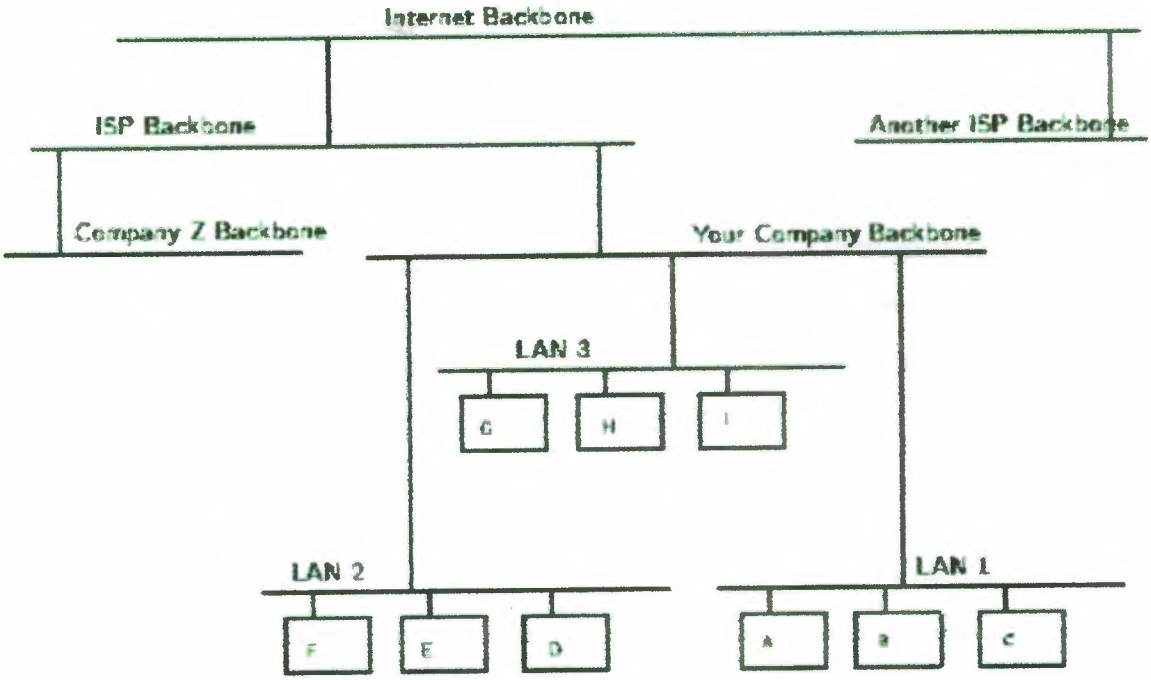

Figure 1.8. A Wider View of Internet-connected Network

The Internet is made up of a wide variety of hosts, from supercomputers to personal computers, including every imaginable type of hardware and software. How do all of these computers understand each other and work together?

1.6. Overview of TCP/IP

TCP/IP (Transport Control Protocol/Internet Protocol) is the language of the Internet. Anything that can learn to speak TCP/IP can play on the Internet. This is functionality that occurs at the Network (IP) and Transport (TCP) layers in the ISO/OSI Reference Model. Consequently, a host that has TCP/IP functionality (such as Unix, OS/2, MacOS, or Windows NT) can easily support applications (such as Netscape's Navigator) that uses the network.

TCP/IP protocols are not used only on the Internet. They are also widely used to build private networks, called internets, that may or may not be connected to the global Internet. An internet that is used exclusively by one organization is sometimes called an intranet

1.6.1 Open Design

One of the most important features of TCP/IP isn't a technological one: The protocol is an open protocol, and anyone who wishes to implement it may do so freely. Engineers and scientists from all over the world participate in the *IETF* (Internet Engineering Task

Force) working groups that design the protocols that make the Internet work. Their time is typically donated by their companies, and the result is work that benefits everyone.

1.6.2 IP

IP is a "network layer" protocol. This is the layer that allows the hosts to actually talk to each other. Such things as carrying datagrams, mapping the Internet address to a physical network address , and routing, which takes care of making sure that all of the devices that have Internet connectivity can find the way to each other.

1.6.3 IP Address

IP addresses are analogous to telephone numbers – when you want to call someone on the telephone, you must first know their telephone number. Similarly, when a computer on the Internet needs to send data to another computer, it must first know its IP address. IP addresses are typically shown as four numbers separated by decimal points, or "dots". For example, 10.24.254.3 and 192.168.62.231 are IP addresses.

If you need to make a telephone call but you only know the person's name, you can look them up in the telephone directory (or call directory services) to get their telephone number. On the Internet, that directory is called the Domain Name System or DNS for short. If you know the name of a server, say www.cert.org, and you type this into your web browser, your computer will then go ask its DNS server what the numeric IP address is that is associated with that name.

1.6.3.1 Static And Dynamic Addressing

Static IP addressing occurs when an ISP permanently assigns one or more IP addresses for each user. These addresses do not change over time. However, if a static address is assigned but not in use, it is effectively wasted. Since ISPs have a limited number of addresses allocated to them, they sometimes need to make more efficient use of their addresses.

Dynamic IP addressing allows the ISP to efficiently utilize their address space. Using dynamic IP addressing, the IP addresses of individual user computers may change over time. If a dynamic address is not in use, it can be automatically reassigned to another computer as needed.

1.6.3.2 Attacks Against IP

A number of attacks against IP are possible. Typically, these exploit the fact that IP does not perform a robust mechanism for *authentication*, which is proving that a packet came from where it claims it did. A packet simply claims to originate from a given address, and there isn't a way to be sure that the host that sent the packet is telling the truth. This isn't necessarily a weakness, *per se*, but it is an important point, because it means that the facility of host authentication has to be provided at a higher layer on the ISO/OSI Reference Model. Today, applications that require strong host authentication (such as cryptographic applications) do this at the application layer.

1.6.3.3 IP Spoofing

This is where one host claims to have the IP address of another. Since many systems (such as router access control lists) define which packets may and which packets may not pass based on the sender's IP address, this is a useful technique to an attacker: he can send packets to a host, perhaps causing it to take some sort of action.

1.6.4 TCP and UDP Ports

TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) are both protocols that use IP. Whereas IP allows two computers to talk to each other across the Internet, TCP and UDP allow individual applications (also known as "services") on those computers to talk to each other.

In the same way that a telephone number or physical mail box might be associated with more than one person, a computer might have multiple applications (e.g. email, file services, web services) running on the same IP address. Ports allow a computer to differentiate services such as email data from web data. A port is simply a number associated with each application that uniquely identifies that service on that computer. Both TCP and UDP use ports to identify services. Some common port numbers are 80 for web (HTTP), 25 for email (SMTP), and 53 for Dmain Name System (DNS).

1.6.5 TCP

TCP is a transport-layer protocol. It needs to sit on top of a network-layer protocol, and was designed to ride atop IP. (Just as IP was designed to carry, among other things, TCP packets.) Because TCP and IP were designed together and wherever you have one, you

typically have the other, the entire suite of Internet protocols are known collectively as TCP/IP. TCP itself has a number of important features that we'll cover briefly.

1.6.5.1 Guaranteed Packet Delivery

Probably the most important is guaranteed packet delivery. Host A sending packets to host B expects to get acknowledgments back for each packet. If B does not send an acknowledgment within a specified amount of time, A will resend the packet.

Applications on host B will expect a data stream from a TCP session to be complete, and in order. As noted, if a packet is missing, it will be resent by A, and if packets arrive out of order, B will arrange them in proper order before passing the data to the requesting application.

This is suited well toward a number of applications, such as a telnet session. A user wants to be sure every keystroke is received by the remote host, and that it gets every packet sent back, even if this means occasional slight delays in responsiveness while a lost packet is resent, or while out-of-order packets are rearranged.

It is not suited well toward other applications, such as streaming audio or video, however. In these, it doesn't really matter if a packet is lost (a lost packet in a stream of 100 won't be distinguishable) but it *does* matter if they arrive late (i.e., because of a host resending a packet presumed lost), since the data stream will be paused while the lost packet is being resent. Once the lost packet is received, it will be put in the proper slot in the data stream, and then passed up to the application.

1.6.6 UDP

UDP (User Datagram Protocol) is a simple transport-layer protocol. It does not provide the same features as TCP, and is thus considered "unreliable". Again, although this is unsuitable for some applications, it does have much more applicability in other applications than the more reliable and robust TCP.

1.6.6.1 Lower Overhead than TCP

One of the things that makes UDP nice is its simplicity. Because it does not need to keep track of the sequence of packets, whether they ever made it to their destination, etc., it has lower overhead than TCP. This is another reason why it's more suited to streaming-data applications: there's less screwing around that needs to be done with making sure all the packets are there, in the right order, and that sort of thing.

1.6.7 Domain Name System (DNS)

DNS is a distributed database system used to match host names with IP addresses. A host normally requests the IP address of a given domain name by sending a UDP message to the DNS server which responds with the IP address or with information about another DNS server.

1.6.8 Telnet

Telnet provides simple terminal access to a host computer. The user is normally authenticated based on user name and password. Both of these are transmitted in plain text over the network however, and is therefore susceptible to capture.

1.6.9 File Transfer Protocols

FTP - The file transfer protocol is one of the most widely and heavily used Internet applications . FTP can be used to transfer both ASCII and binary files. Separate channels are used for commands and data transfer. Anonymous FTP allows external users to retrieve files from a restricted area without prior arrangement or authorisation. By convention users log in with the userid "anonymous" to use this service. Some sites request that the user's electronic mail address be used as the password.

CHAPTER TWO

CRYPTOGRAPHY SYSTEMS

2.1 Introduction

The origin of the word cryptology lies in ancient Greek. The word cryptology is made up of two components: "kryptos", which means hidden and "logos" which means word. Cryptology is as old as writing itself, and has been used for thousands of years to safeguard military and diplomatic communications. For example, the famous Roman emperor Julius Caesar used a cipher to protect the messages to his troops. Within the field of cryptology one can see two separate divisions: cryptography and cryptanalysis. The cryptographer seeks methods to ensure the safety and security of conversations while the cryptanalyst tries to undo the farmer's work by breaking his systems.

The main goals of modern cryptography can be seen as: user authentication, data authentication (data integrity and data origin authentication), non-repudiation of origin, and data confidentiality. In the following section we will elaborate more on these services. Subsequently we will explain how these services can be realized using cryptographic primitives.

A cryptographic system (or a cipher system) is a method of hiding data so that only certain people can view it. Cryptography is the practice of creating and using cryptographic systems. Cryptanalysis is the science of analyzing and reverse engineering cryptographic systems. The original data is called plaintext. The protected data is called cipher text. Encryption is a procedure to convert plaintext into cipher text. Decryption is a procedure to convert cipher text into plaintext. A cryptographic system typically consists of algorithms, keys, and key management facilities. There are two basic types of cryptographic systems: symmetric ("private key") and asymmetric ("public key").

Symmetric key systems require both the sender and the recipient to have the same key. This key is used by the sender to encrypt the data, and again by the recipient to decrypt the data. Key exchange is clearly a problem. How do you securely send a key that will enable you to send other data securely? If a private key is intercepted or stolen, the adversary can act as either party and view all data and communications. You can think of the symmetric crypto system as akin to the Chubb type of door locks. You must be in possession of a key to both open and lock the door. Asymmetric cryptographic systems are considered much more flexible. Each user has both a public key and a private key. Messages are encrypted with one key and can be decrypted only by the other key. The

public key can be published widely while the private key is kept secret. If Alice wishes to send Bob a secret, she finds and verifies Bob's public key, encrypts her message with it, and mails it off to Bob. When Bob gets the message, he uses his private key to decrypt it. Verification of public keys is an important step. Failure to verify that the public key really does belong to Bob leaves open the possibility that Alice is using a key whose associated private key is in the hands of an enemy. Public Key Infrastructures or PKI's deal with this problem by providing certification authorities that sign keys by a supposedly trusted party and make them available for download or verification. Asymmetric ciphers are much slower than their symmetric counterparts and key sizes are generally much larger. You can think of a public key system as akin to a Yale type door lock. Anyone can push the door locked, but you must be in possession of the correct key to open the door.

2.2 Encryption and Decryption

Data that can be read and understood without any special measures is called *plaintext* or *clear text.* The method of disguising plaintext in such a way as to hide its substance is called *encryption.* Encrypting plaintext results in unreadable gibberish called *cipher text.* You use encryption to ensure that information is hidden from anyone for whom it is not intended, even those who can see the encrypted data. The process of reverting ciphertext to its original plaintext is called *decryption.* Figure 2.1 illustrates this process.

Figure 2.1 Encryption and decryption

2.3 How Does Cryptography Work?

A *cryptographic algorithm,* or cipher, is a mathematical function used in the encryption and decryption process. A cryptographic algorithm works in combination with a *key* — a word, number, or phrase to encrypt the plaintext. The same plaintext encrypts to different cipher text with different keys. The security of encrypted data is entirely dependent on two things: the strength of the cryptographic algorithm and the secrecy of

the key. A cryptographic algorithm, plus all possible keys and all the protocols that make it work comprise a *cryptosystem.* PGP is a cryptosystem.

2.4 Public Key Cryptography

The problems of key distribution are solved by *public key cryptography,* the concept of which was introduced by Whitfield Diffie and Martin Hellman in 1975. (There is now evidence that the British Secret Service invented it a few years before Diffie and Hellman, but kept it a military secret — and did nothing with it. [J H Ellis: The Possibility of Secure Non-Secret Digital Encryption, CESG Report, January 1970])Public key cryptography is an asymmetric scheme that uses a *pair* of keys for encryption: a *public key,* which encrypts data, and a corresponding *private,* or *secret key* for decryption. You publish your public key to the world while keeping your private key secret. Anyone with a copy of your public key can then encrypt information that only you can read. Even people you have never met.

It is computationally infeasible to deduce the private key from the public key. Anyone who has a public key can encrypt information but cannot decrypt it. Only the person who has the corresponding private key can decrypt the information.

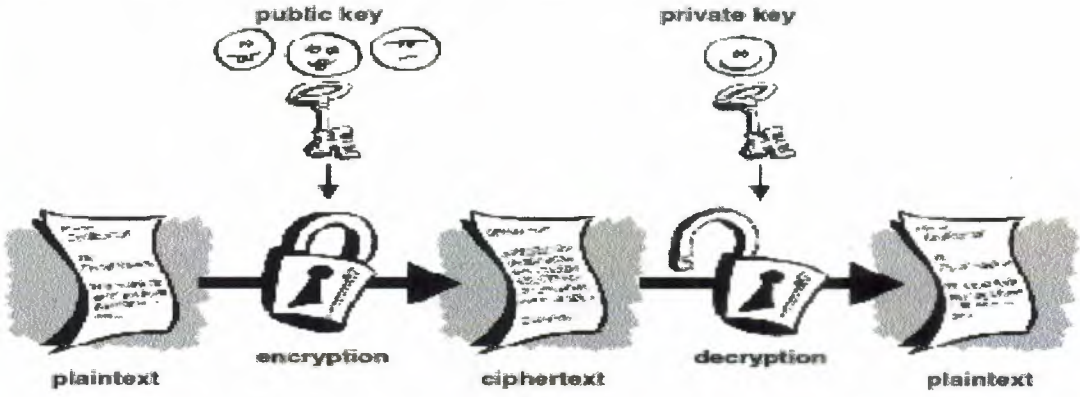

Figure 2.2 Public key encryption

The primary benefit of public key cryptography is that it allows people who have no preexisting security arrangement to exchange messages securely. The need for sender and receiver to share secret keys via some secure channel is eliminated; all communications involve only public keys, and no private key is ever transmitted or shared.

2.5 Cryptographic Services

2.5.1 User Authentication

If you log to a computer system there must (or at least should) be some way that you can convince it of your identity. Once it knows your identity, it can verify whether you are entitled to enter the system. The same principal applies when one person tries to communicate with another: as a first step you want to verify that you are communicating with the right person. Therefore there must be some way in which you can prove your identity. This process is called user authentication. There are several ways to obtain user authentication. You can give him something only you can know: a password, a (predestined) user-id, a pin code, and so on. Or you could have some specific items with which you can identify yourself: a magnetic strip card, a smart card (a hand-held computer the size of a credit-card), a token. One might make use of biometric properties; it is a well-known fact that fingerprints, the shape of the hand and retinal pattern of a person are good decision criteria. These however require specialized equipment and thus a big investment. However, these biometric systems are not perfect: some legitimate users will inevitably fail the identification and some intruders will be accepted as genuine. Other techniques include measurements of how a person types his name or writes his signature, or can take into account the location of the user.

Figure 2.3 User authentication

For the time being the first two methods are the ones generally applied, and many practical systems use a combination of both. Since the user's memory is limited, this information should not vary too much over time. Whether it is a password, a pin code or a user-id, all these items are being defined at a certain time and often don't change from there on. One might argue that you could change your password, but this is not done each time you access the computer. This indicates that someone who can eavesdrop this information will later be able to impersonate the user. A similar observation holds true

for a magnetic strip card or memory chip. All these systems provide static authentication only.

If the user possesses a device which can perform simple computations, the security can be increased significantly by introducing the well-known challenge-response idea. If a person tries to identify himself to the system, the system generates a random challenge and sends it to the person or to his device. In case of a token (a mini-calculator), the user will have to enter the challenge on the keyboard. The device will then compute the corresponding response, using secret information which has been assigned to him. This response is then sent back to the system, which verifies it (see figure 2.3). If more sophisticated protocols are used, the verifier does not need secret information (this requires public-key protocols), or will even not learn the secret of the users (this requires zero-knowledge protocols). Note that in this case the procedure does not authenticate the user but rather his device. In order to increase the security, the user should authenticate him with respect to the device, using something he alone knows. This makes the device useless if it is stolen. In general, one also requires that the computer authenticates itself to the person logging on. If both parties are authenticated to each other, we use the term mutual authentication.

2.5.2 Data Authentication

Data authentication consists of two components: the fact that data has not been modified (data integrity) and the fact that you know who the sender is (data origin authentication).

2.5.3 Data Integrity

A data integrity service guarantees that the content of the message, that was sent, has not been tampered with. Data integrity by itself is not meaningful: it does not help you to know that the data you have received has not been modified, unless you know it has been sent directly to you by the right person. Therefore it should always be combined with data origin authentication.

You should always be alert for possible intruders in your network or in your communication system. A well-known example is the Internet that connects universities and companies world-wide. Electronic mail over the Internet does not offer any security. As a consequence, an educated computer user can tap into the messages that

are being transmitted over the line. It is very easy to read and modify someones electronic mail, which is commonly seen as being private.

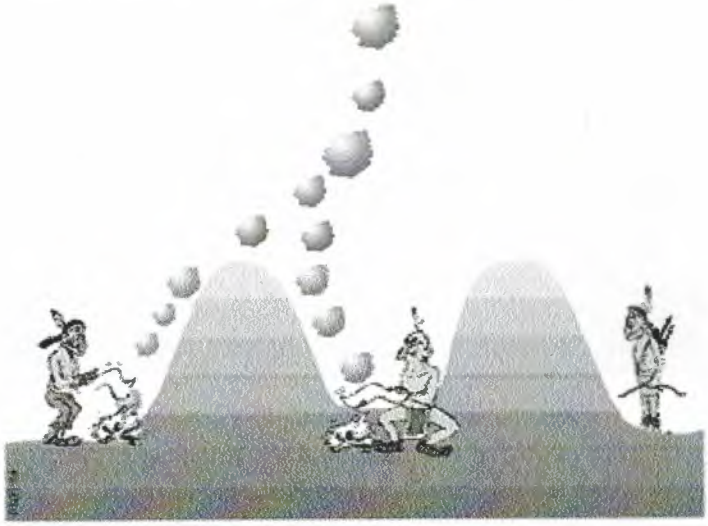

Figure 2.4 Data integrity

In general, we take the point of view of figure 2.4 we have a (lice) that sends a message to B (ob). There is also an enemy who taps the line between them. If you don't support data integrity, this enemy can just change the message and then relay it to B. B will not see that the message has been tampered with and will assume A really intended it the way he got it. One could argue that active wire-tapping is difficult. In general wire-tapping is only a matter of cost: tapping a telephone line is obviously easier than tapping a coaxial cable or a micro-wave. Active wire-taps (modifying and then relaying the messages) are also more difficult than passive wire-taps (listening in on the messages).

2.5.4 Data Origin Authentication

Here one wants to make sure that the person who is claiming to be the sender of the message really is the one from whom it originates. In figure 2.5, if A sends a message to B, but the enemy intercepts it and sends it to B, claiming A has sent it, how can B be sure of the real origin of this data? A variation on this theme is: the enemy could send a message to B claiming it A is the originator. Thanks to cryptography, there are techniques to ensure against this type of fraud.

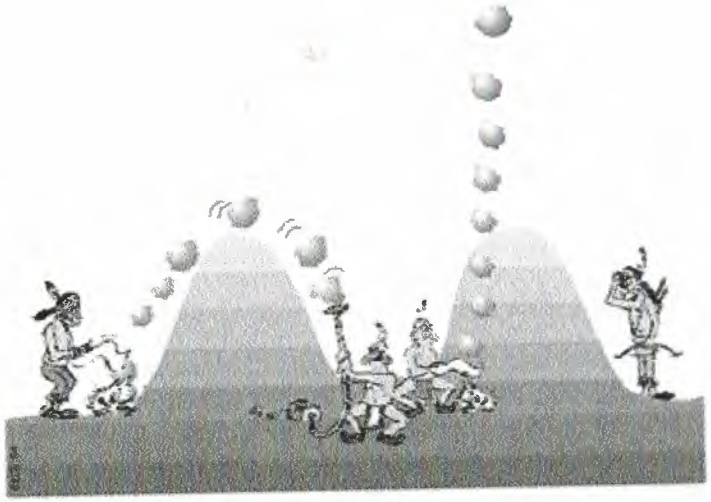

Figure 2.5 Data origin authentication

2.5.5 Non-Repudiation of Origin

Non-repudiation protects against denial by one of the entities involved in a communication of having participated in all or part of the communication. Non-repudiation with proof of origin protects against any attempts by the sender to repudiate having sent a message, while non-repudiation with proof of delivery protects against any attempt by the recipient to deny, falsely, having received a message.

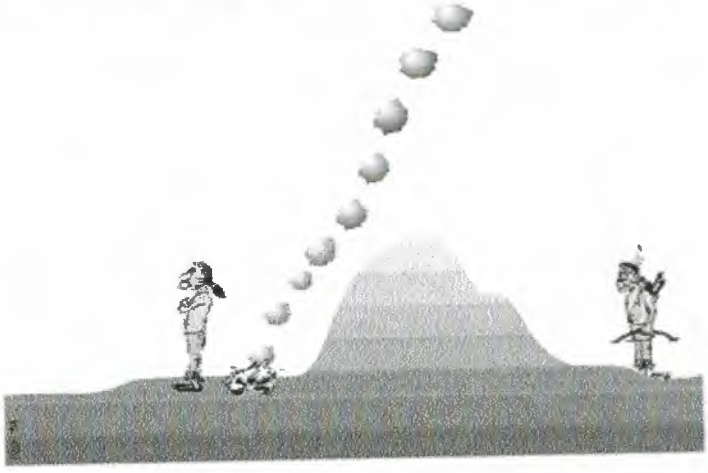

Figure 2.6 Non-repudiation of origin

An example will illustrate the importance of non-repudiation of origin. Suppose B is the owner of a mail-order company and he decides to let his customers order through electronic mail. For him it is really important that he can show to an arbitrary third party that A really ordered the things he is claiming otherwise it would be easy for a customer to deny the purchase of the goods (see figure 2.6). In a paper and pencil world, non-repudiation is provided by a manual signature.

2.5.6 Data Confidentiality

This aspect of data security certainly is the oldest and best known. The example of Caesars cipher given in the introduction clearly demonstrates this. The fact that confidentiality was considered to be much more important than authentication of both sender and data, together with non-repudiation of origin can be explained as follows: the latter services have been provided implicitly by the physical properties of the channel: a letter was written in a recognizable handwriting, with a seal and a signature.

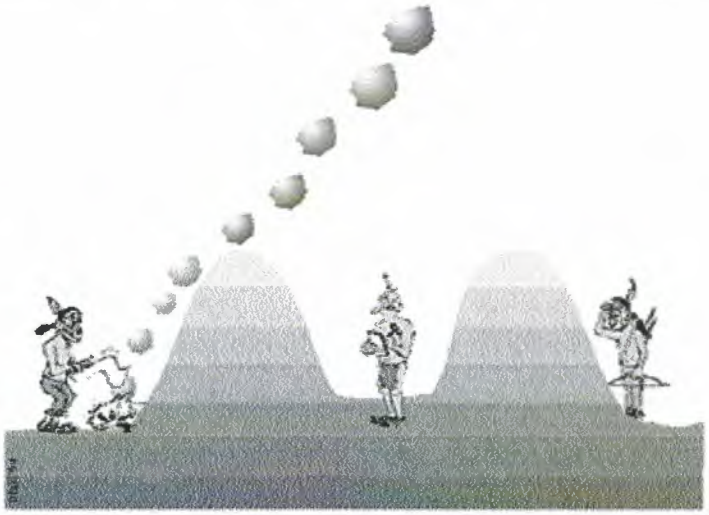

Figure 2.7 Data confidentiality

With data confidentiality we try to protect ourselves against unauthorized disclosure of the message. Referring to figure 2.7, if a sends a message to B, but the enemy intercepts it, one wants to make sure that this enemy never understands his contents. Confidentiality protection is very important in the medical world and also in the banking sector. World-wide there are several million transactions each day and all of these have to be passed from one financial institution to another. If there were no way to protect confidentiality, everybody would be able to see who had purchased what, who has made what kind of withdrawal, and so on.

Clearly this would violate individuals and companies rights to privacy. In order to provide confidentiality, it is necessary to transform the message with a cipher.

2.6 Cryptographic Primitives

The above cryptographic services can be realized by several cryptographic primitives: we distinguish between primitives for encryption, primitives for authentication, and cryptographic protocols. Encryption primitives can be used to provide confidentiality,

authentication primitives can be used to provide data authentication. We will also discuss protocols for user authentication and for key management.

Figure 2.8 Encryption

2.6.1 Encryption Primitives

In cryptography one often makes use of encryption. With encryption we transform the clear text (or plaintext) into cipher text. To get back to the original text, we apply the inverse transformation, called decryption. These transformations themselves are public: this makes it possible to analyze these algorithms and to develop efficient implementations. However they use a secret parameter: the keys which are known only by the sender and/or the receiver. This key is the only thing one needs to know in order to encipher or decipher. Thus it is really important to manage one's keys and keep them secret where necessary. We discuss two types of encryption primitives, symmetric or conventional ciphers and asymmetric or public-key ciphers.

2.6.2 Symmetric Ciphers

Basically there are two kinds of encryption-schemes. The oldest ones and most used until now are the symmetric ciphers. In these schemes, the key used to decipher the cipher text is equal to the one used to encipher the plaintext.

The best known cipher in this category is the Data Encryption Standard (DES) that was adopted in 1977 by the American NBS (National Bureau of Standards) as FIPS 46. Since then it has been used all over the world and until now no major flaws have been discovered. Symmetric cryptography uses a single private key to both encrypt and decrypt data. Any party that has the key can use it to encrypt and decrypt data. They are also referred to as block ciphers. Symmetric cryptography algorithms are typically fast and are suitable for processing large streams of data. The disadvantage of symmetric

cryptography is that it presumes two parties have agreed on a key and been able to exchange that key in a secure manner prior to communication. This is a significant challenge. Symmetric algorithms are usually mixed with public key algorithms to obtain a blend of security and speed. The DES makes use of a 56-bit key which is unfortunately short. Researchers have estimated that exhaustively searching for all possible values of this key in 1 day will currently require an investment of about US$ 200,000 (this requires only a few pairs of plaintext and corresponding cipher text). During the last years E. Biham and A. Shamir, and later M. Matsui have published attacks which break DES in the academic sense (i.e., they require significantly less operations), though this is no threat to the DES in practice, since they require huge amounts of known or chose plaintexts respectively.

Better security can be achieved using the triple-DES. In this way, we effectively obtain a key of 112 bits and this is sufficiently large. At the same time, one is protected against further improvements in academic attacks on the DES.

It is not sufficient to choose a secure cipher; one also has to specify a secure mode of operation. Depending on the nature of the communication channel or storage space, one will choose between Cipher-Block-Chaing (CBC), Cipher-Feedback (CFB), and Output-Feedback (OFB) (as specified in FIPS 81). Encryption block by block (or Electronic Code Book (ECB) mode) will only be used for encryption of keys.

2.6.3 Asymmetric Ciphers

The asymmetric or public-key ciphers are the most recent cryptographic tools. In contrary to the symmetric systems the key used to encipher and the one used to decipher are different. Each partner thus has two keys. He keeps one key secret and makes the other one public. If A wants to send a message to B, he just enciphers it with B's public key. Since B is the only one who has access to the secret key, B is the only one who can decipher the message and read the contents.

Figure 2.9 Key management with asymmetric cipher

The most popular public-key cipher is the RSA system (RSA stands for Rivest, Shamir and Adleman, the names of the three inventors). The security of this scheme is related to the mathematical problem of factorization: it is easy to generate two large primes and to multiply them, but given a large number that is the product of two primes, it requires a huge amount of computation to find the two prime factors. At the moment of writing of this text, the biggest number that has been factorized was about 430 bits long and attacks on numbers of 512 bits have been announced for 1997. Therefore the absolute minimum length of the key in the RSA system has to be set at 640 bits; 768 or 1024 bits are required for any system that requires security for more than a few months.

2.6.4 Symmetric Versus Asymmetric Ciphers

The biggest drawback of the asymmetric systems up until now has been the relative low performance compared to the symmetric ones. For example, the implementation of DES on a 586/90 PC could achieve a 15 Mbit/s encryption rate while the RSA implementation on the same PC has only a 6 Kbits/s rate. Thus as you can see the DES is typically 1000 times faster than the RSA-scheme.

Public-key cryptography is also called asymmetric. It uses a secret key that must be kept from unauthorized users and a public key that can be made public to anyone. Both the public key and the private key are mathematically linked; data encrypted with the public key can be decrypted only by the private key, and data signed with the private key can only be verified with the public key.

The public key can be published to anyone. Both keys are unique to the communication session. Public-key cryptographic algorithms use a fixed buffer size. Private-key

cryptographic algorithms use a variable length buffer. Public-key algorithms cannot be used to chain data together into streams like private-key algorithms can. With private-key algorithms only a small block size can be processed, typically 8 or 16 bytes. Public-key systems provide significant benefits in terms of key management: if every user generates his own key, only an authentic channel is required, eliminating (expensive) secret channels like couriers.

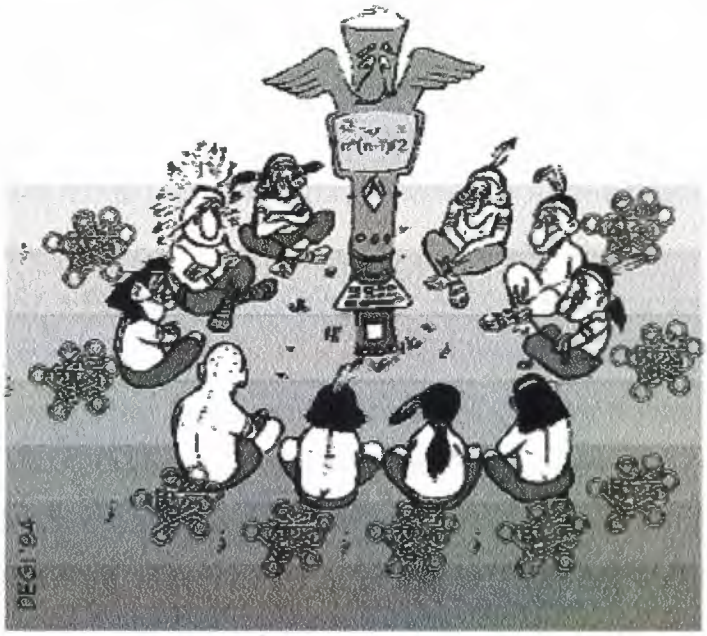

Figure 2.10 Key Management with Asymmetric Cipher

In systems without a central trusted server, the number of keys can be reduced. Indeed, suppose we have a network of n users each of whom wanting to communicate with the others. Since each communication requires a secret key, the total number of keys required equals n*(n-1)/2. In the public-key system each user only needs a personal public/secret key pair, yielding a total of only 2n keys. If n equals 1000 this would mean 2000 versus 499500. In systems with a central management system, both approaches require the same number of keys. However, the central system can be off-line in the case of use of public key technology, which reduces the cost and minimizes the security risks.

In practice one thus often encounters hybrid systems in which one uses a public-key system for the distribution of the secret keys and a symmetric cipher for the bulk encryption of the data.

2.7 Authentication Primitives

2.7.1 One-Way Functions and Hash Codes

A one-way function is defined as a function f such that for every x in the domain of f, f(x) is easy to compute; but for virtually all y in the range of f, it is computationally infeasible to find an x such that y=f(x). In addition one requires that it is hard to find a second pre-image: given an x and the corresponding value of f(x), it should be hard to find an x' different from x which has the same image under f.

One-way functions are used to protect passwords: one will store a one-way image of the password in the computer rather than the password itself. One applies then the one-way function to the input of the user and verifies whether the outcome agrees with the value stored in the table.

A hash function is a function which maps an input of arbitrary length into a fixed number of output bits. In order to be useful for cryptographic applications, a hash function has to satisfy some additional requirements. One can distinguish two types of hash functions. A MAC (Message Authentication Code) that uses a secret key, and an MDC (Manipulation Detection Code) that works without a key. For a MAC one requires that it should be impossible to compute the MAC without knowledge of the secret key. For an MDC one requires that it is a one-way function, and - in most cases - that it is collision resistant, which means that it should be hard to find two arguments hashing to the same result.

Hash functions can be used to protect the authenticity of large quantities of data with a short secret key (MAC), or to protect the authenticity of a short string (MDC). Sometimes an MDC is used in combination with encryption, which can yield protection of both confidentiality and authenticity. Hash algorithms are one-way mathematical algorithms that take an arbitrary length input and produce a fixed length output string. A hash value is a unique and extremely compact numerical representation of a piece of data. MD5 produces 128 bits for instance. It is computationally improbable to find two distinct inputs that hash to the same value (or ``collide''). Hash functions have some very useful applications. They allow a party to prove they know something without revealing what it is, and hence are seeing widespread use in password schemes. They can also be used in digital signatures and integrity protection. There are several other types of cryptographic algorithms like elliptic curve and stream ciphers. For a complete and thorough tutorial on implementing cryptographic systems we suggest ``Applied Cryptography'' by Bruce Schneier. There are several schemes which have been

proposed for use as hash functions. The widely used construction for a MAC is the CBC mode of the DES (with an additional output transformation), as specified in ISO-9797. Several MDC's have been constructed based on the DES. Other dedicated designs are SHA (Secure Hash Algorithm or FIPS 180), and RIPE-MD 160. These hash functions achieve a very high throughput (Mbit/s), even in software implementations.

2.7.2 Digital Signature

Public-key techniques can also be used for other purposes than for enciphering information. If Alice adds some redundancy to her message and transforms the result using her secret key, anyone who knows Alice's public key can verify that this message was sent by Alice (by verifying the redundancy). In this way one can create a digital signature, which is the equivalent of the hand-written signature on a document. Public-key and private-key algorithms can also be used to form digital signatures. Digital signatures authenticate the identity of a sender (if you trust the sender's public key) and protect the integrity of data. You may also hear the term MAC (Message Authentication Code). Since it is not physically connected to the signed data or the originator, it will depend on this data and on the secret key of the originator. Several signature schemes have been proposed. The RSA public-key cryptosystem is the only one which can be used for both enciphering and digital signatures. Schemes which can only be used for digital signature purposes are the DSA and the Fiat-Shamir scheme.

Note that it is possible to produce a digital signature based on conventional ciphers like the DES. However, these schemes are less efficient in terms of memory and computations. Other constructions use a conventional cipher in combination with tamper resistant hardware: this offers only a limited protection.

Assume Bob has received from Alice a digitally signed message. If Alice subsequently denies having sent the message, Bob can go to a third party (e.g., a judge), who will be able to obtain Alice's public key. Subsequently he can verify the validity of the signature. In this way a digital signature can provide non-repudiation of origin. It is easy to see that it provides in addition data authentication, i.e., data integrity and data origin authentication.

2.7.3 Hash Functions versus Digital Signatures

Hash functions can only be used in a situation where the parties mutually trust each other: they cannot be used to resolve a dispute (unless one uses, in addition tamper

resistant hardware).As in the case of encryption, hash functions tend to be three orders of magnitude faster than digital signatures. This explains why in general one will first compute the hash code of the message with a fast hash function and subsequently apply the digital signature to this short hash code. This provides digital signatures which are not only faster and shorter, but also more secure.

2.8 Cryptographic Protocols

A cryptographic protocol is an interaction between one or more entities to achieve a certain goal. In fact, encryption and digital signatures can be seen as a special case of cryptographic protocols.

While a huge number of protocols have been developed, we will restrict this section to two types of protocols: protocols for user authentication and protocols for key management.

2.8.1 User Authentication Protocols

The design of cryptographic protocols for user authentication is very complex. A large number of protocols have been presented in the available literature, many of which exhibit some weaknesses. The simplest protocol providing unilateral authentication consists of sending a password. More complex challenge-response protocols can be designed in which the user does not transmit his secret information. They are based on an encryption algorithm, a MAC or a digital signature and the use, in addition, of so called nonce's (never used more than once) : random numbers, sequence numbers or time stamps. More complex protocols are required to achieve mutual authentication.

2.8.2 Key Management Protocols

On e of the main links in the cryptographic keychain is the key management protocol: every cryptographic service will make use of cryptographic keying material, whose confidentiality and/or integrity has to be protected. For the distribution of this keying material, one can use a new cryptographic primitive, and ultimately, a physical channel. In this way one builds a key hierarchy: secret keys for bulk encryption with a symmetric cipher system will be encrypted using an asymmetric cipher system and signed with a digital signature scheme. The public keys of the asymmetric cipher can be distributed via an authentic channel which can be provided for example by combining conventional

mail with voice authentication. An alternative is to sign these public keys with a single master key: now one only has to distribute a single master key via an authentic channel. These signed public keys are called certificates. The central authority certifies that a certain public key belongs to a particular user. The commonly used scheme nowadays in based on the ITU-T X.509 recommendation.

Note that there also exist public-key protocols which result in the agreement of a secret key between two parties, by exchanging public keys or parameters. A well known example in this class is the Diffie-Hellman key agreement scheme. This protocol is different from a key transport protocol, in which one party generates the secret key and enciphers it with the public key of the other party. The key agreement protocols have the advantage that they result in an increased security level.

In the context of public-key cryptography, revocation of public keys is very important: once the user's secret key is compromised, anybody can read his messages or forge his signatures. Although public-key systems require no on-line central management system, the system has to provide a means to protect the user in the case by warning the other users that his public key is no longer valid.

CHAPTER THREE

NETWORK SECURITY

3.1 Introduction

The process of protecting data and equipment from unauthorized access is collectively known as network security. The importance of implementing good network security procedures is highlighted when you consider the ramifications of not taking such precautions: data can be accidentally or intentionally erased from the system; a competitor can gain an unfair advantage by accessing confidential data; and the use of network resources can be lost, yielding a corresponding loss of productivity.

It is the role of network administration to take preventive action to ensure that the risk of such losses is minimized. However, care must be taken to balance the reduction of security risks against the ensuing loss in ease of use and availability of the networked systems. Security procedures and system flexibility are diametrically opposed concepts. Every step taken by a network administrator to prevent unauthorized access creates another step that an authorized user must take to gain access to the data. It is important to analyze each system on a network and place appropriate security restrictions on an individual basis.

3.2 Security Risks

The first step to understanding security is to know what the potential risks are, or more specifically, to determine the type and level of security risks for the company. Security risks are unique to each organization because they are dependent on the nature of the business and the environment in which the company operates. For example, the security risks for a high profile dot com company that solely operates on the Internet will be very different from a small manufacturing company that does little on the Web.

Security risk is determined by identifying the assets that need to be protected. The assets could include customer credit card information, proprietary product formulas, employee data, the company's Web site, or other assets that are deemed to be important to the organization. Once the assets are identified, the next step is to determine the criticality of the assets to the company. For example, if the asset is considered to be very important to the company, then the level of security for that asset should be high.

The next step is assessing the likelihood of a potential attack. While security measures must always be put in place to protect the assets of the company, the risks increase as the probability of an attack rises. For example, it is more likely for an outside intruder to attempt to break into a Web site selling consumer goods than a small manufacturing company making rubber bands. Therefore, while both companies must have security measures, the company with the Web site must deploy a higher level of security. Now that the process of determining security risk has been defined, some of the more common security risks are briefly discussed below.

3.3 Security Threats

The first step in evaluating security risks is to determine the threats to system security. Although the term network security has been commonly categorized as protecting data and system resources from infiltration by third-party invaders, most security breeches are initiated by personnel inside the organization. Organizations will spend hundreds of thousands of dollars on securing sensitive data from outside attack while taking little or no action to prevent access to the same data from unauthorized personnel within the organization.

The threat from hackers has been largely overstated. Individuals who fit into this group have more of a Robin Hood mentality than a destructive mentality. Most hackers, or crackers as they prefer to be called, are more interested in the thrill of breaking into the system than they are in causing damage once they succeed in gaining access. Unfortunately, there is an increasing trend for hackers to be employed by other entities as an instrument to gain access to systems.

As the amount of critical data stored on networked systems has increased, the appeal of gaining access to competitors' systems has also increased. In highly competitive industry segments, an entire underground market exists in the buying and trading of product and sales data. By gaining access to research and development information from a competitor, millions of dollars and years of research can be eliminated.

Another external threat is that of government intrusion, both from the domestic government and from foreign governments. Agencies such as the Federal Bureau of Investigation and the Internal Revenue Service can have vested interests in gaining access to critical tax and

related information. Foreign governments are especially interested in information that could represent an economic or national defense advantage

3.3.1 Types and Sources of Network Threats

First of all, we will get into the types of threats there are against networked computers, and then some things that can be done to protect yourself against various threats.

3.3.1.1 Denial of Service

DoS (Denial-of-Service) attacks are probably the nastiest, and most difficult to address. These are the nastiest, because they're very easy to launch, difficult (sometimes impossible) to track, and it isn't easy to refuse the requests of the attacker, without also refusing legitimate requests for service.

The premise of a DoS attack is simple: send more requests to the machine than it can handle. There are toolkits available in the underground community that make this a simple matter of running a program and telling it which host to blast with requests. The attacker's program simply makes a connection on some service port, perhaps forging the packet's header information that says where the packet came from, and then dropping the connection. If the host is able to answer 20 requests per second, and the attacker is sending 50 per second, obviously the host will be unable to service all of the attacker's requests, much less any legitimate requests (hits on the web site running there, for example).

Such attacks were fairly common in late 1996 and early 1997, but are now becoming less popular.

Some things that can be done to reduce the risk of being stung by a denial of service attack include

Not running your visible-to-the-world servers at a level too close to capacity using packet filtering to prevent obviously forged packets from entering into your network address space.

Obviously forged packets would include those that claim to come from your own hosts, addresses reserved for private networks as defined in RFC 1918 [4], and the *loopback* network (127.0.0.0).

3.3.1.2 Unauthorized Access

Unauthorized access is a very high-level term that can refer to a number of different sorts of attacks. The goal of these attacks is to access some resource that your machine should not provide the attacker. For example, a host might be a web server, and should provide anyone with requested web pages. However, that host should not provide command shell access without being sure that the person making such a request is someone who should get it, such as a local administrator.

3.3.1.2.1 Executing Commands Illicitly

It is obviously undesirable for an unknown and untrusted person to be able to execute commands on your server machines. There are two main classifications of the severity of this problem: normal user access, and administrator access. A normal user can do a number of things on a system (such as read files, mail them to other people, etc.) that an attacker should not be able to do. This might, then, be all the access that an attacker needs. On the other hand, an attacker might wish to make configuration changes to a host (perhaps changing its IP address, putting a start-up script in place to cause the machine to shut down every time it's started or something similar). In this case, the attacker will need to gain administrator privileges on the host.

3.3.1.2.2 Confidentiality Breaches

We need to examine the threat model: what is it that you're trying to protect yourself against? There is certain information that could be quite damaging if it fell into the hands of a competitor, an enemy, or the public. In these cases, it's possible that compromise of a normal user's account on the machine can be enough to cause damage (perhaps in the form of PR, or obtaining information that can be used against the company, etc.)

While many of the perpetrators of these sorts of break-ins are merely thrill-seekers interested in nothing more than to see a shell prompt for your computer on their screen, there are those who are more malicious, as we'll consider next. (Additionally, keep in mind that it's possible that someone who is normally interested in nothing more than the thrill could be persuaded

3.3.2 Where Do They Come From?

How, though, does an attacker gain access to your equipment? Through any connection that you have to the outside world. This includes Internet connections, dial-up modems, and even physical access. How do you know that one of the temps that you've brought in to help with the data entry isn't really a system cracker looking for passwords, data phone numbers, vulnerabilities and anything else that can get him access to your equipment?

In order to be able to adequately address security, all possible avenues of entry must be identified and evaluated. The security of that entry point must be consistent with your stated policy on acceptable risk levels.

3.4 Security Concepts and Technology

This section includes a brief description of network security concepts and technology. This information can be used to understand some of the security methods that are deployed throughout the network.

A comprehensive security approach requires that the company's different levels of management collectively create an enterprise-wide security approach by determining the appropriate security policies. The business side of the company typically uses policies to manage, so this concept is not unfamiliar to managers. A security policy defines the assets that need to be protected, who can have access to those assets, when they can have access, and how they are allowed to use the assets.

More important is the fact that the managers who own certain corporate assets (for example, customer or company data) have excellent insight into what policies should be designed to control access to these assets. Therefore, input from these managers is invaluable for securing the assets of the company. Unfortunately, there are many companies today where the security management of the company is totally in the hands of IT staff. While they are very knowledgeable and competent, the IT staff must have input from the owners of the data to truly provide a comprehensive and consistent security management strategy. Without this input, the security of the company's assets may be at risk.

Once the high-level security policies have been determined, the security strategy can be developed from them. The security strategy should include a

security plan that defines the tools and technologies to be used, and how they should be deployed. In addition, more specific access policies can be developed.

The security plan should include strategies that secure the perimeter of the enterprise, as well as strategies to secure the internal network. While the perimeter defense is a necessary piece of a complete security approach, the security strategy should not end there. Once intruders have access to the internal network, there must be security measures to prevent them from causing irreparable damage. A combination of security tools and technologies must be deployed throughout the network to ensure a secure network.

3.4.1 Firewalls

The concept of the firewall is much like the walled cities of medieval times, where an external perimeter was constructed to keep intruders out and to protect the residents within. The gates are designed both to control the entry of outsiders and to allow residents to leave the walled city. In addition, the gates provide limited-access points that are more easily defended against intruders.

Originally, many companies viewed firewalls as solid walls that would totally block outside entry to the enterprise. However, with the increased popularity of the Internet and the interactions of e-business, that approach is no longer acceptable. Administrators must now strike a balance between allowing required services through the firewall, while ensuring the security of the company's assets. As a result, the role of the firewall has evolved from being a solid perimeter wall to becoming the gates in the enterprise's perimeter wall.

A number of terms specific to firewalls and networking are going to be used throughout this section, so let's introduce them all together.

3.4.1.1 Bastion Host

A general-purpose computer used to control access between the internal (private) network (intranet) and the Internet (or any other untrusted network). Typically, these are hosts running a flavor of the Unix operating system that has been customized in order to reduce its functionality to only what is necessary in order to support its functions. Many of the general-purpose features have been turned off, and in many cases, completely removed, in order to improve the security of the machine.

3.4.1.2Access Control List (ACL).

Many routers now have the ability to selectively perform their duties, based on a number of facts about a packet that comes to it. This includes things like origination address, destination address, destination service port, and so on. These can be employed to limit the sorts of packets that are allowed to come in and go out of a given network.

3.4.1.3 Demilitarized Zone (DMZ)

The DMZ is a critical part of a firewall: it is a network that is neither part of the untrusted network, nor part of the trusted network. But, this is a network that connects the untrusted to the trusted. The importance of a DMZ is tremendous: someone who breaks into your network from the Internet should have to get through several layers in order to successfully do so. Those layers are provided by various components within the DMZ.

3.4.1.4 Proxy

This is the process of having one host act in behalf of another. A host that has the ability to fetch documents from the Internet might be configured as a *proxy server* , and host on the intranet might be configured to be *proxy clients* . In this situation, when a host on the intranet wishes to fetch the web page, for example, the browser will make a connection to the proxy server, and request the given URL. The proxy server will fetch the document, and return the result to the client. In this way, all hosts on the intranet are able to access resources on the Internet without having the ability to direct talk to the Internet.

Now that the concept of firewalls has been described, it would be useful to have a basic understanding of how they work. The traffic coming into or going out of the corporate network originates from a location that is identified with an IP address (a unique network address). In addition, the traffic is composed of services that may be required by the enterprise, such as e-mail, File Transfer Protocol (FTP), Telnet, and many others. When setting up a firewall, the security administrator must define what services are to be allowed (both inbound and outbound), and whether to filter incoming and outgoing traffic based on IP addresses. The techniques that most firewalls use to filter incoming and outgoing traffic to the corporate networks are IP filtering, a proxy, or a combination of both methods.

3.4.1.5 IP Filtering

Every device on a TCP/IP network (the Internet, for example) is identified by a unique IP address. IP filtering is an access-control mechanism that filters network traffic based on IP addresses and requested services. It does this by using access control lists (ACLs), of which there are two types:

Host-based access control lists, which describe the services that are allowed or denied for each host or network. Service-based access lists, which describe the hosts or networks that are allowed or denied to use each service.

The firewall will reject any services or hosts that are denied access in the ACLs. Likewise, it will accept services from hosts that are allowed access in the ACLs. Network devices, such as firewalls and routers, can use ACLs to control access. In a recent Enterprise Management Associates study on security, 50% of the 100 respondents polled reported that they use IP filtering. Of those respondents that use IP filtering, 86% of them use IP filtering on their firewalls.

ACL is almost like a guest list at an exclusive and high-security event. The list contains the names of those "guests" who have been invited and are allowed to attend the event. In addition, the guest list may also list services, such as the caterer, florist, or entertainers, who should be allowed to enter. The guest list may even name specific people who were not invited, and request that the security staff be especially vigilant to prevent them from entering. It may also include instructions that certain services, such as the media, should not be allowed to enter. So the ACL acts like a guest list by naming who can and cannot have access, in addition to describing services that can and cannot have access through the firewall or router.

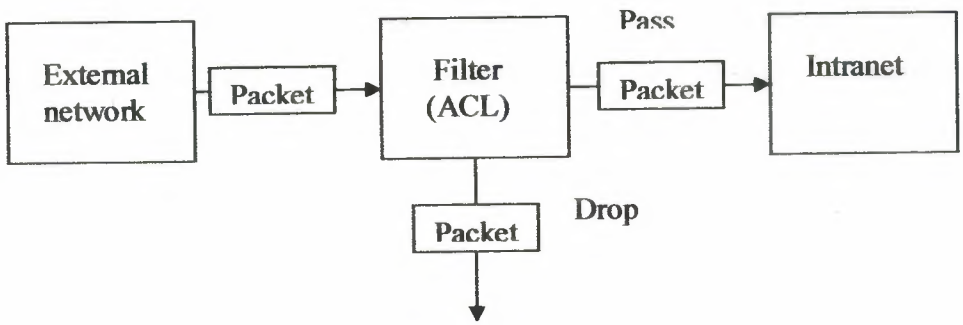

Figure 3.1. IP Filtering

To be effective, access control lists must be carefully and comprehensively constructed to ensure that unauthorized access and services are not allowed into the network. The ordering of the rules in the ACL is important because the first match that the firewall finds is executed. Creating and maintaining comprehensive ACLs can be a tedious task for security administrators of large and complex networks, especially if the definitions of ACLs are done manually. Because manually managing ACLs throughout the enterprise is difficult, in some cases only bare minimum ACLs are used, or they are not as widely deployed as they should be. To take full advantage of the benefits that IP filtering can offer, security administrations need to use ACL management tools that facilitate easy deployment and administration of ACLs.

IP filtering provides flexibility, allowing administrators to create both simple access rules and a sophisticated set of rules to define what traffic will be allowed to pass through the firewall. In addition, IP filtering is a relatively fast method for controlling access because it is typically processed in the system kernel.

3.5 Secure Network Devices

It's important to remember that the firewall only one entry point to your network. Modems, if you allow them to answer incoming calls, can provide an easy means for an attacker to sneak *around* (rather than *through*) your front door (or, firewall). Just as castles weren't built with moats only in the front, your network needs to be protected at all of its entry points.

3.5.1 Secure Modems (Dial-Back Systems)

If modem access is to be provided, this should be guarded carefully. The *terminal server* , or network device that provides dial-up access to your network needs to be actively administered, and its logs need to be examined for strange behavior. Its password need to be strong -- not ones that can be guessed. Accounts that aren't actively used should be disabled. In short, it's the easiest way to get into your network from remote: guard it carefully.

There are some remote access systems that have the feature of a two-part procedure to establish a connection. The first part is the remote user dialing into the system, and

providing the correct userid and password. The system will then drop the connection, and call the authenticated user back at a known telephone number. Once the remote user's system answers that call, the connection is established, and the user is on the network. This works well for folks working at home, but can be problematic for users wishing to dial in from hotel rooms and such when on business trips.

Other possibilities include one-time password schemes, where the user enters his userid, and is presented with a ``challenge," a string of between six and eight numbers. He types this challenge into a small device that he carries with him that looks like a calculator. He then presses enter, and a ``response" is displayed on the LCD screen. The user types the response, and if all is correct, he login will proceed. These are useful devices for solving the problem of good passwords, without requiring dial-back access. However, these have their own problems, as they require the user to carry them, and they must be tracked, much like building and office keys.

No doubt many other schemes exist. Take a look at your options, and find out how what the vendors have to offer will help you *enforce your security policy effectively.*

3.5.2 Virtual Private Networks (NPN)

Given the ubiquity of the Internet, and the considerable expense in private leased lines, many organizations have been building *VPNs* (Virtual Private Networks). Traditionally, for an organization to provide connectivity between a main office and a satellite one, an expensive data line had to be leased in order to provide direct connectivity between the two offices. Now, a solution that is often more economical is to provide both offices connectivity to the Internet. Then, using the Internet as the medium, the two offices can communicate.

The danger in doing this, of course, is that there is no privacy on this channel, and it's difficult to provide the other office access to ``internal" resources without providing those resources to everyone on the Internet.

VPNs provide the ability for two offices to communicate with each other in such a way that it looks like they're directly connected over a private leased line. The session between them, although going over the Internet, is private (because the link is encrypted), and the link is

convenient, because each can see each others' internal resources without showing them off to the entire world.

A number of firewall vendors are including the ability to build VPNs in their offerings, either directly with their base product, or as an add-on. If you have need to connect several offices together, this might very well be the best way to do it.

VPNs are a viable way to use the ubiquitous public Internet to securely transmit private data between sites. It is a lower cost solution to traditional dedicated connections.

CHAPTER FORE

F IREWALLS DISCRIBTION AND STRUCTRE

4.1 Overview

The last topic that we consider is the firewall. Firewalls can be an effective means of protecting a local system or network of systems from network- based security threats while at the same time affording access to the outside world via wide area networks and the Internet. We begin this chapter with an overview of the functionality and design principles of firewalls. Next, we address the issue of the security of the firewall itself and, in particular, the concept of a trusted system, or secure operating system.

4.2 Firewall Design Principles

Information systems in corporations, government agencies, and other organizations have undergone a steady evolution:

• Centralized data processing system, with a central mainframe supporting a number of directly connected terminals

• Local area network (LAN) interconnecting PCs and terminals to each other and the mainframe

• Premises network, consisting of a number of LANs, interconnecting PCs, servers, and perhaps a mainframe or two

• Enterprise-Wide network, consisting of multiple, geographically distributed premises networks interconnected by a private wide area network (WAN)

• Internet connectivity, in which the various premises networks all hook into the Internet and may or may not also be connected by a private WAN. Internet connectivity is no longer an option for most organizations. The information and services available are essential to the organization. Moreover, individual users want and need Internet access, and if this is not provided via their LAN, they will use dial-up capability from their PC to an Internet service provider (ISP). However, while Internet access provides benefits to the organization, it enables the outside world to reach and interact with local network assets. This creates a threat to the organization. While it is possible to equip each workstation and server on the premises network with strong security features, such as intrusion protection, this is not a practical approach. Consider a network with hundreds or even thousands of systems, running a mix of various versions of UNIX, plus Windows 95, 98, and NT. When a security flaw is discovered, each potentially affected system

must be upgraded to fix that flaw. The alternative, increasingly accepted, is the firewall. The firewall is inserted between the premises network and the Internet to establish a controlled link and to erect an outer security wall or perimeter. The aim of this perimeter is to protect the premises network from Internet-based attacks and to provide a single choke point where security and audit can be imposed. The firewall may be a single computer system or a set of two or more systems that cooperate to perform the firewall function.

In this section, we look first at the general characteristics of firewalls. Then we look at the types of firewalls currently in common use. Finally, we examine some of the most common firewall configurations.

4.3 Firewall Characteristics

Lists the following design goals for a firewall:

1. All traffic from inside to outside, and vice versa, must pass through the firewall. This is achieved by physically blocking all access to the local network except via the firewall. Various configurations are possible, as explained later; in this section.

2. Only authorized traffic, as defined by the local security policy, will be allowed to pass. Various types of firewalls are used, which implement various types of security policies, as explained later in this section.

3. The firewall itself is immune to penetration. This implies that use of a trusted system with a secure operating system.

Originally, firewalls focused primarily on service control, but they have since evolved to provide all four:

Service control: Determines the types of Internet services that can be accessed, inbound or outbound. The firewall may filter traffic on the basis of IP address and TCP port number; may provide proxy software that receives and interprets each service request before passing it on; or may host the server software itself, such as a Web or mail service.

• **Direction control:** Determines the direction in which particular service requests may be initiated and allowed to flow through the firewall.

• **User control:** Controls access to a service according to which user is attempting to access it. This feature is typically applied to users inside the firewall perimeter (local users). It may also be applied to incoming traffic from external users; the latter requires some form of secure

authentication technology.

• **Behavior control:** Controls how particular services are used. For example, the firewall may filter e-mail to eliminate spam, or it may enable external access to only a portion of the information on a local Web server. Before proceeding to the details of firewall types and configurations, it is best to summarize what one can expect from a firewall. The following capabilities are within the scope of a firewall:

1. A firewall defines a single choke point that keeps unauthorized users out of the protected network, prohibits potentially vulnerable services from entering or leaving the network, and provides protection from various kinds of IP spoofing and routing attacks. The use of a single choke point simplifies security management since security capabilities are consolidated on a single system or set of systems.

2. A firewall provides a location for monitoring security-related events. Audits and alarms can be implemented on the firewall system.

3. A firewall is a convenient platform for several Internet functions that are not security related. These include a network address translator, which maps local addresses to Internet addresses, and a network management function that audits or logs Internet usage.

4. A firewall can serve as the platform for Insect. Using the tunnel mode capability the firewall can be used to implement virtual private networks.

Firewalls have their limitations, including the following:

1. The firewall cannot protect against attacks that bypass the firewall. Internal systems may have dial-out capability to connect to an ISP. An internal LAN may support a modem pool that provides dial-in capability for traveling employees and telecommuters.

2. The firewall does not protect against internal threats, such as a disgruntled employee or an employee who unwittingly cooperates with an external attacker.

3. The firewall cannot protect against the transfer of virus-infected programs or files. Because of the variety of operating systems and applications supported inside the perimeter, it would be impractical and perhaps impossible for the firewall to scan all incoming files, e-mail, and messages for viruses.

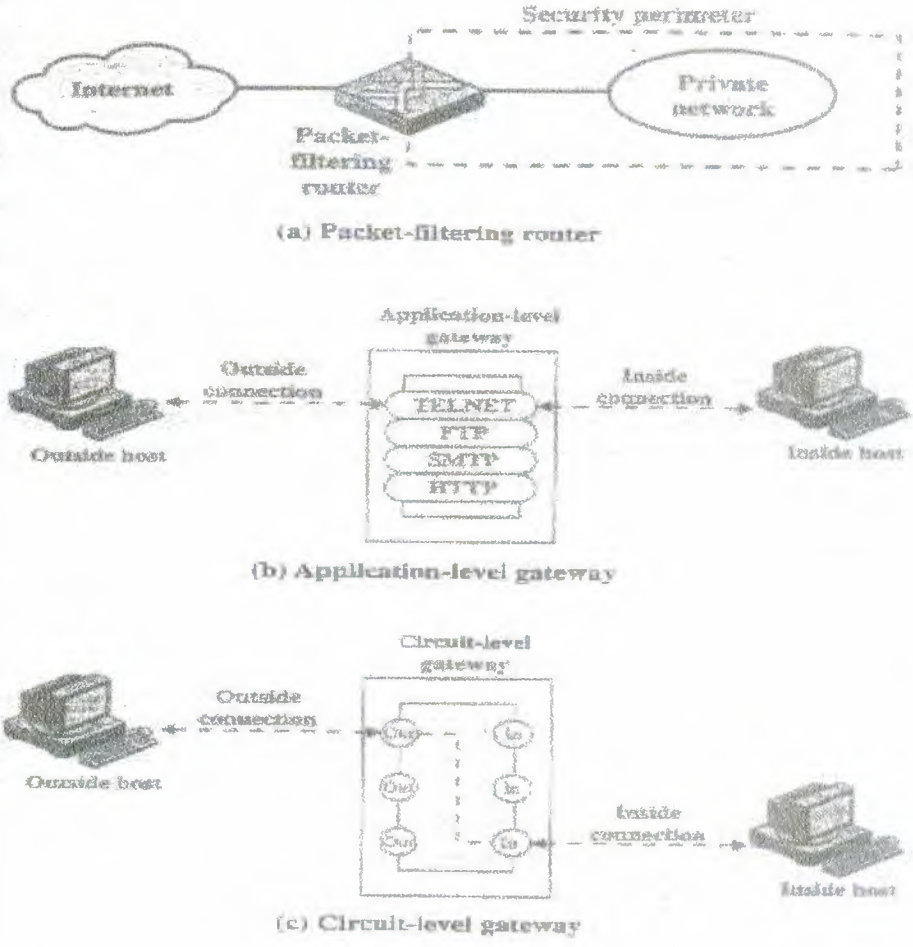

Figure 4.1 Firewall Types

4.4 Types of Firewalls

Figure 4.1 based on figures in [SEM1E96], illustrates the three common types of firewalls: packet filters, application-level gateways, and circuit-level gateways. We examine each of these in turn.

4.4.1 Packet-Filtering Router

A packet-filtering router applies a set of rules to each incoming IF packet and then forwards or discards the packet. The router is typically configured to filter packets going in both directions (from and to the internal network). Filtering rules are based on fields in the IP and transport (e.g., TCP or UDP) header, including source and destination IP address, IP protocol field (which

defines the transport protocol), and TCP or UDP port number (which defines an application such as SNMP or TELNET).

The packet filter is typically set up as a list of rules based on matches to fields in the IF or TCP header. If there is a match to one of the rules, that rule is invoked to determine whether to forward or discard the packet. If there is no match to any rule, than a default action is taken. Two default policies are possible:

- **Default** = discard: That which is not expressly permitted is prohibited.
- **Default** = forward: That which is not expressly prohibited is permitted.

The default discard policy is the more conservative. Initially, everything is blocked, and services must be added on a case-by-case basis. This policy is more visible to users, who are more likely to see the firewall as a hindrance. The default forward policy increases ease of use for end users but provides reduced security; the security administrator must, in essence, react to each new security threat as it becomes known.

Table 4.1, from [BELL94I, gives some examples of packet-filtering rule sets. In each set, the rules are applied top to bottom.

Table 4.1 Packet-Filtering Examples

A.

action	ourhost	port	theirhost	port	comment
block	*	*	SPIGOT	*	we don't trust these people
allow	OUR-GW	25	*	*	connection to our SMTP port

B.

action	ourhost	port	theirhost	port	comment
block	*	*	*	*	default

C.

action	ourhost	port	theirhost	port	comment
allow	*	*	*	25	connection to their SMTP port

D.

action	src	port	dest	port	flags	comment
allow	{our hosts}	*	*	25		our packets to their SMTP port
allow	*	25	*	*	ACK	their replies

E.

action	src	port	dest	port	flags	comment
allow	{our hosts}	*	*	*		our outgoing calls
allow	*	*	*	*	ACK	replies to our calls
allow	*	*	*	>1024		traffic to nonservers

A. Inbound mail is allowed (port 25 is for SMTP incoming), but only to a gateway host. However, mail from a particular external host, SPIGOT, is blocked because that host has a history of sending massive files in e-mail messages.

B. This is an explicit statement of the default policy. All rule sets include this rule implicitly as the last rule.

C. This rule set is intended to specify that any inside host can send mail to the outside. A TCP packet with a destination port of *25* is routed to the SMTP server on the destination machine. The problem with this rule is that the use of port 25 for SMTP receipt is only a default; an outside machine could be configured to have some other application linked to port 25. As this rule is written, an attacker could gain access to internal machines by sending packets with a **TCP** source port number of 25.

D. This rule set achieves the intended result that was not achieved in C. The rules take advantage of a feature of **TCP** connections. Once a connection is set up, the **ACK** flag of a **TCP** segment is

set to acknowledge segments sent from the other side. Thus, this rule set states that it allows **IP** packets where the source IP address is one of a list of designated internal hosts and the destination TCP port number is 25. It also allows incoming packets with a source port number of 25 that include the ACK flag in the TCP segment. Note that we explicitly designate source and destination systems to define these rules explicitly.

E. This rule set is one approach to handling FTP connections. With FTP, two TCP connections are used: a control connection to set up the file transfer and a data connection for the actual file transfer. The data connection uses a different port number that is dynamically assigned for the transfer. Most servers, and hence most attack targets. Live on low-numbered ports: most outgoing calls tend to use a higher-numbered port, typically above 1023. Thus, this rule set allows

- Packets that originate internally
- Reply packets to a connection initiated by an internal machine
- Packets destined for a high-numbered port on an internal machine.

This scheme requires that the systems be configured so that only the appropriate numbers are in use.

Rule set E points out the difficulty in dealing with applications at the packet- filtering level. Another way to deal with FTP and similar applications is an application-level gateway, described later in this section.

One advantage of a packet-filtering router is its simplicity. Also, packet filters typically are transparent to users and are very fast. The disadvantages include the difficulty of setting up packet filter rules correctly and the lack of authentication.

[SEME96] lists some of the attacks that can be made on packet-filtering routers and the appropriate countermeasures:

• **IP addresses spoofing:** The intruder transmits packets from the outside with a source IF address field containing an address of an internal host. The attacker hopes that the use of a spoofed address will allow penetration of systems that employ simple source address security, in which packets from specific trusted internal hosts are accepted. The countermeasure is to discard packets with an inside source address if the packet arrives on an external interface.

• **Source routing attacks:** The source station specifies the route that a packet should take as it crosses the Internet, in the hopes that this will bypass security measures that do not analyze the source routing information. The countermeasure is to discard all packets that use this option.

• **Tiny fragment attacks:** The intruder uses the IP fragmentation option to create extremely small fragments and force the TCP header information into a separate packet fragment. This attack is designed to circumvent filtering rules that depend on TCP header information. The attacker hopes that only the first fragment is examined by the filtering router and that the remaining fragments are passed through. A tiny fragment attack can be defeated by discarding all packets where the protocol type is TCP and the IP Fragment Offset is equal to 1.

4.4.2 Application-Level Gateway

An application-level gateway, also called a proxy server, acts as a relay of application-level traffic (Figure 4.lb). The user contacts the gateway using a TCP/IP application, such as Telnet or FTP, and the gateway asks the user for the name of the remote host to be accessed. When the user responds and provides a valid user ID and authentication information, the gateway contacts the application on the remote host and relays TCP segments containing the application data between the two endpoints. If the gateway does not implement the proxy code for a specific application, the service is not supported and cannot be forwarded across the firewall. Further, the gateway can be configured to support only specific features of an application that the network administrator considers acceptable while denying all other features.

Application-level gateways tend to be more secure than packet filters. Rather than trying to deal with the numerous possible combinations that are to be allowed and forbidden at the TCP and IP level, the application-level gateway need only scrutinize a few allowable applications. In addition, it is easy to log and audit all incoming traffic at the application level.

A prime disadvantage of this type of gateway is the additional processing overhead on each connection. In effect. There are two spliced connections between the end users, with the gateway at the splice point, and the gateway must examine and forward all traffic in both directions.

4.4.3 Circuit-Level Gateway

A third type of firewall is the circuit-level gateway (Figure 4.lc). This can be a stand-alone system or it can be a specialized function performed by an application-level gateway for certain applications. A circuit-level gateway does not permit an end-to-end TCP connection; rather, the gateway sets up two TCP connections, one between itself and a TCP user on an inner host and one between itself and a TCP user on an outside host. Once the two connections are established,

the gateway typically relays TCP segments from one connection to the other without examining the contents. The security function consists of determining which connections will be allowed.
A typical use of circuit-level gateways is a situation in which the system administrator trusts the internal users. The gateway can be configured to support application-level or proxy service on inbound connections and circuit-level functions for outbound connections. In this configuration, the gateway can incur the processing overhead of examining incoming application data for forbidden functions but does not incur that overhead on outgoing data.
An example of a circuit-level gateway implementation is the SOCKS package [KOBL92]; version 5 of SOCKS is defined in RFC 1928. The RFC defines SOCKS in the following fashion:

The protocol described here is designed to provide a framework for client-server applications in both the TCP and UDP domains to conveniently and securely use the services of a network firewall. The protocol is conceptually a "shim-layer" between the application layer and the transport layer, and such as does not provide network- layer gateway services, such as forwarding of ICMP massages.

SOCKS consist of the following components:
• The SOCKS server, which runs on a UNIX-based firewall.
• The SOCKS client library, which runs on internal hosts protected by the firewall.
• SOCKS-ivied versions of several standard client programs such as FTP and TELNET. The implementation of the SOCKS protocol typically involves the recompilation or relining of TCP-based client applications to use the appropriate encapsulation routines in the SOCKS library.

When a TCP-based client wishes to establish a connection to an object that is reachable only via a firewall (such determination is left up to the implementation), it must open a TCP connection to the appropriate SOCKS port on the SOCKS server system. The SOCKS service is located on TCP port 1080. If the connection request succeeds, the client enters a ne2otiation for the authentication method to be used, authenticates with the chosen method, and then sends a relay request. The SOCKS server evaluates the request and either establishes the appropriate connection or denies it. UDP exchanges are handled in a similar fashion. In essence, a TCP connection is opened to authenticate a user to send and receive UDP segments, and the UDP segments are

forwarded as long as the TCP connection is open.

4.4.4 Bastion Host

A bastion host is a system identified by the firewall administrator as a critical strong point in the network's security. Typically, the bastion host serves as a platform for an application-level or circuit-level gateway. The following are common characteristics of a bastion host:

• The bastion host hardware platform executes a secure version of its operating system, making it a trusted system.

• Only the services that the network administrator considers essential are installed on the bastion host. These include proxy applications such as Telnet, DNS, FTP, SMTP, and user authentication.

• The bastion host may require additional authentication before a user is allowed access to the proxy services. In addition, each proxy service may require its own authentication before granting user access.

• Each proxy is configured to support only a subset of the standard application's command set.

• Each proxy is configured to allow access only to specific host systems. This means that the limited command/feature set may be applied only to a subset of systems on the protected network.

• Each proxy maintains detailed audit information by logging all traffic, each connection, and the duration of each connection. The audit log is an essential tool for discovering and terminating intruder attacks.

• Each proxy module is a very small software package specifically designed for network security. Because of its relative simplicity, it is easier to check such modules for security flaws.

For example, a typical UNIX mail application may contain over 20,000 lines of code, while a mail proxy may contain fewer than 1000 [SEMIE96].

• Each proxy is independent of other proxies on the bastion host. If there is a problem with the operation of any proxy, or if a future vulnerability is discovered, it can be uninstalled without affecting the operation of the other proxy applications. Also if the user population requires support for a new service, the network administrator can easily install the required proxy on the bastion host.

• A proxy generally performs no disk access other than to read its initial configuration file. This

makes it difficult for an intruder to install Trojan horse sniffers or other dangerous files on the bastion host.

• Each proxy runs as a no privileged user in a private and secured directory on the bastion host.

4.5 Firewall Configurations

In addition to the use of a simple configuration consisting of a single system, such as a single packet-filtering router or a single gateway (Figure 4.1), more complex configurations are possible and indeed more common. Figure 4.2, based on figures in [SEMIE96], illustrates three common firewall configurations. We examine each of these in turn.

In the **screened host firewall. Single-homed bastion** configuration (Figure 4.1) the firewall consists of two systems: a packet-filtering router and a bastion host. Typically, the router is configured so that

1. For traffic from the Internet, only IP packets destined for the bastion frost are allowed in.
2. For traffic from the internal network, only IP packets from the bastion host are allowed out.

The bastion host performs authentication and proxy functions. This configuration has greater security than simply a packet-filtering router or an application-level gateway alone, for two reasons. First, this configuration implements both packet-level and application-level filtering, allowing for considerable flexibility in defining security policy. Second, an intruder must generally penetrate two separate systems before the security of the internal network is compromised.

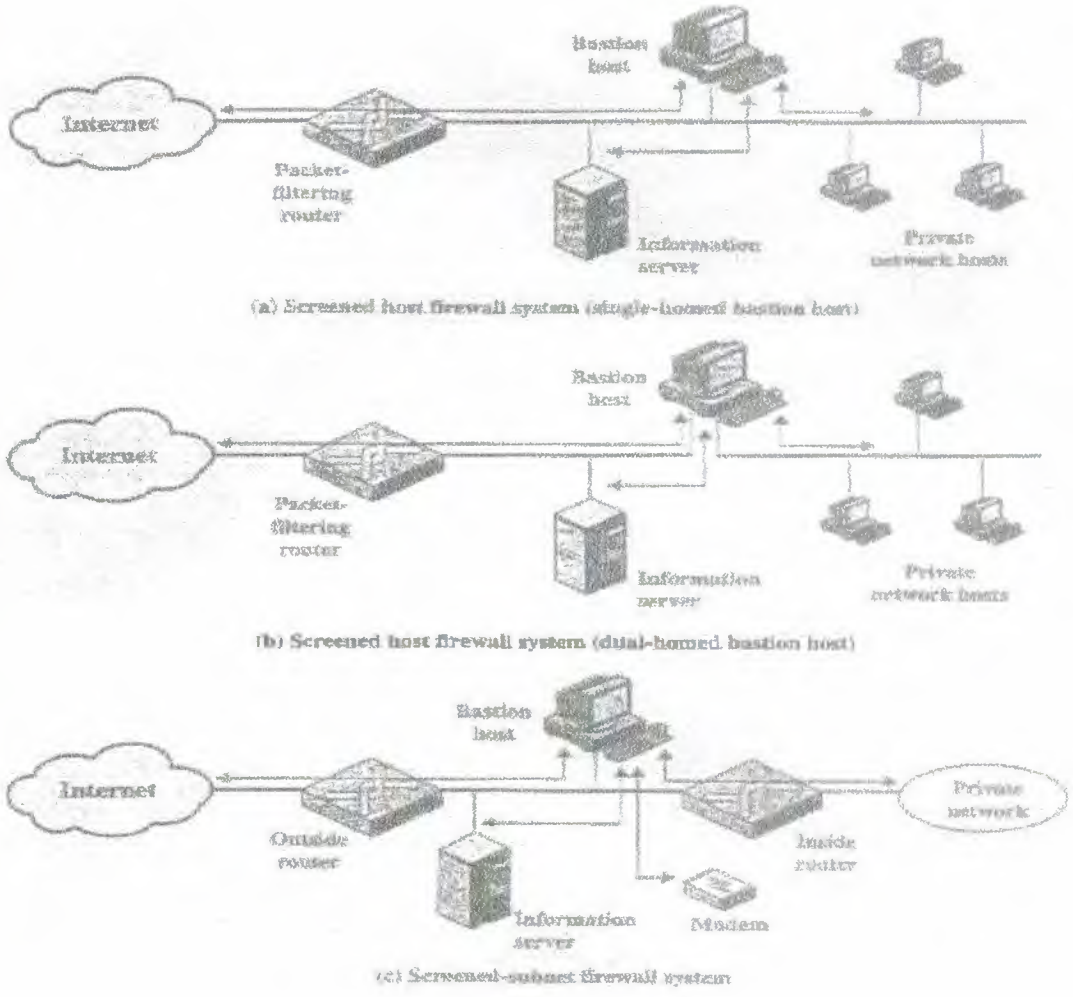

Figure 4.2 Firewall Configurations.

This configuration also affords flexibility in providing direct Internet access. For example, the internal network may include a public information server, such, as a Web server, for which a high level of security is not required. In that case, the router can be configured to allow direct traffic between the information server and the Internet.

In the single-homed configuration just described, if the packet-filtering router is completely compromised, traffic could flow directly through the router between the Internet and other hosts on the private network. The screened **host firewall,** dual-homed **bastion** configuration physically prevents such a security breach (Figure 4.2b). The advantages of dual layers of security that were present in the previous configuration are present here as well. Again, an information server or

other hosts can be allowed direct communication with the router if this is in accord with the security policy.

The screened subnet firewall configuration of Figure 4.2c is the most secure **of** those **we have considered. In this** configuration, two packet-filtering routers **are used, one** between the bastion host and the Internet and one between the bastion host **and** the internal network. This configuration creates **an** isolated sub network, which may consist of simply the bastion host, but may also include one or more information servers and modems for dial-in capability. Typically, both the Internet and the internal network have access to hosts on the screened subnet, but traffic across the screened subnet is blocked. This configuration offers several advantages:

• There are now three levels of defense to thwart intruders.

• The outside router advertises only the existence of the screened subnet to the Internet; therefore, the internal network is invisible to the Internet.

• Similarly, the inside router advertises only the existence of the screened subnet to the internal network; therefore, the systems on the inside network cannot construct direct routes to the Internet.

4.6 Trusted System

One way to enhance the ability of a system to defend against intruders and malicious programs is to implement trusted system technology. This section provides a brief overview of this topic. We begin by looking at some basic concepts of data access control.

4.6.1 Data Access Control

Following successful logon, the user has been granted access to one or a set of hosts and applications. This is generally not sufficient for a system that includes sensitive data in its database. Through the user access control procedure, a user can be identified to the system. Associated with each user, there can be a profile that specifies permissible operations and file accesses. The operating system can then enforce rules based on the user profile. The database management system, however, must control access to specific records or even portions of records. For example, it may be permissible for anyone in administration to obtain a list of company personnel, but only selected individuals may have access to salary information. The issue is more than just one of level of detail. Whereas the operating system may grant a user

permission to access a file or use an application, following which there are no further security checks, the database management system must make a decision on each individual access attempt. That decision will depend not only on the user's identity but also on the specific parts of the data being accessed and even on the information already divulged to the user.

A general model of access control as exercised by a file or database management system is that of an access matrix (Figure 4.3a). The basic elements of the model are as follows:

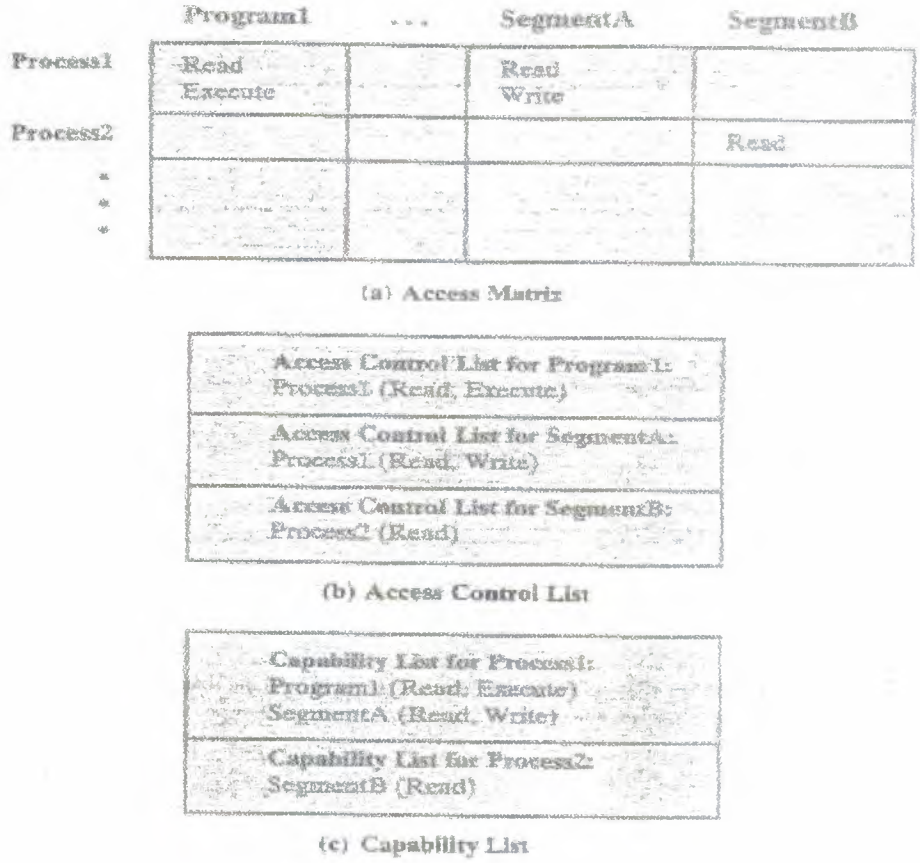

Figure 4.3 Access Control Structures.

• **Subject:** An entity capable of accessing objects. Generally, the concept of subject equates with that of process. Any user or application actually gains access to an object by means of a process that represents that user or application.

• **Object:** Anything to which access is controlled. Examples include *files,* portions of files,

programs, and segments of memory.

• **Access right:** The way in which an object is accessed by a subject. Examples are read, write, and execute.

One axis of the matrix consists of identified subjects that may attempt data access. Typically, this list will consist of individual users or user groups, although access could be controlled for terminals, hosts, or applications instead of or in addition to users. The other axis lists the objects that may be accessed. At the greatest level of detail, objects may be individual data fields. More aggregate groupings, such as records, files, or even the entire database, may also be objects in the matrix. Each entry in the matrix indicates the access rights of that subject for that object.

In practice, an access matrix is usually sparse and is implemented by decomposition in one of two ways. The matrix may be decomposed by columns, yielding **access control lists** (Figure 4.3b). Thus, for each object, an access control list lists users and their permitted access rights. The access control list may contain a default, or public, entry. This allows users that are not explicitly listed as having special rights to have a default set of rights. Elements of the list may include individual users as well as groups of users.

Decomposition by rows yields **capability tickets** (Figure 4.3c). A capability ticket specifies authorized objects and operations for a user. Each user has a number of tickets and may be authorized to loan or give them to others. Because tickets may be dispersed around the system, they present a greater security problem than access control lists. In particular, the ticket must be unforgivable. One way to accomplish this is to have the operating system hold all tickets on behalf of users. These tickets would have to be held in a region of memory inaccessible to users.

4.6.2 The Concept of Trusted Systems

Much of what we have discussed so far has been concerned with protecting a given message or item from passive or active attack by a given user. A somewhat different but widely applicable requirement is to protect data or resources on the basis of levels of security. This is commonly found in the military, where information is categorized as unclassified (U), confidential (C), secret (S), top secret (TS), or beyond. This concept is equally applicable in other areas, where information can be organized into gross categories and users can be granted clearances to access certain categories of data. For example, the highest level of security might be for strategic

corporate planning documents and data, accessible by only corporate officers and their staff; next might come sensitive financial and personnel data, accessible only by administration personnel, corporate officers, and so on.

When multiple categories or levels of data are defined, the requirement is referred to as **multilevel security.** The general statement of the requirement for multilevel security is that a subject at a high level may not convey information to a subject at a lower or noncom parable level unless that flow accurately reflects the will of an authorized user. For implementation purposes, this requirement is in two parts an~ is simply stated. A multilevel secure system must enforce

• No **read up:** A subject can only read an object of less or equal security level.

This is referred to in the literature as the **simple security** property.

• **No write down:** A subject can only write into an object of greater or equal security level. This is referred to in the literature as the **Property** (pronounced star property).

These two rules, if properly enforced, provide multilevel security. For a data processing system, the approach that has been taken, and has been the object of much research and development, is based on the reference monitor concept.

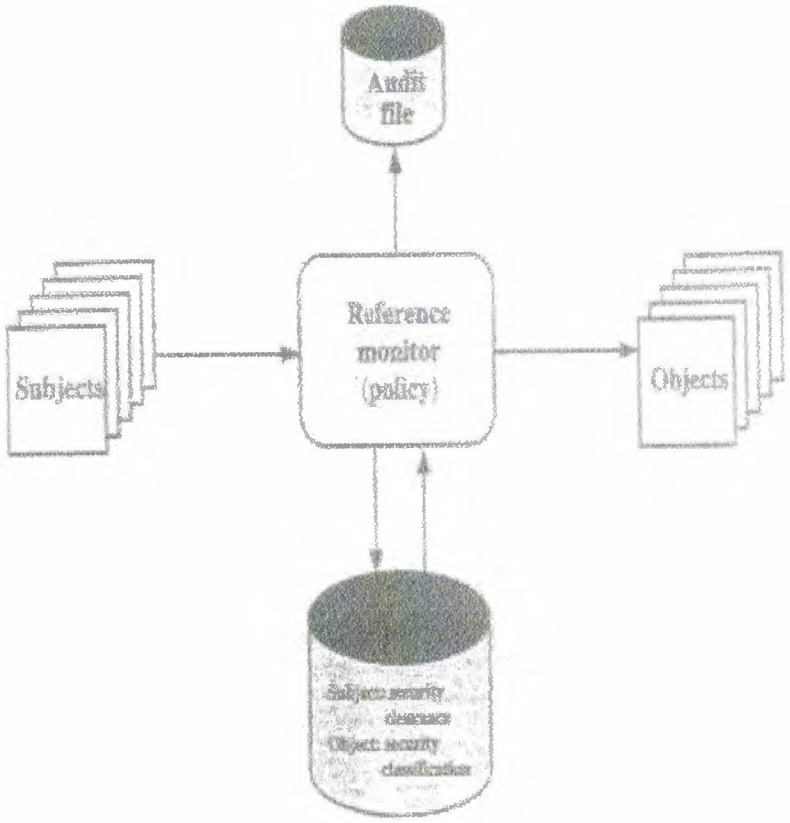

Figure 4.4 Reference Monitor Concepts.

This approach is depicted in Figure 4.4. The reference monitor is a controlling element in the hardware and operating system of a computer that regulates the access of subjects to objects on the basis of security parameters of the subject and object. The reference monitor has access to a file, known as the *security* kernel database that lists the access privileges (security clearance) of each subject and the protection attributes (classification level) of each object. The reference monitor enforces the security rules (no read up, a write down) and has the following properties:

- **Complete mediation:** The security rules are enforced on every access, not just, for example, when a file is opened.
- **Isolation:** The reference monitor and database are protected from unauthorized modification.
- **Verifiability:** The reference monitor's correctness must be provable. That is, it must be possible to demonstrate mathematically that the reference monitor enforces the security rules and provides complete mediation and isolation.

These are stiff requirements. The requirement for complete mediation means that every access to data within main memory and on disk and tape must be mediated. Pure software implementations impose too high a performance penalty to be practical; the solution must be at least partly in hardware. The requirement for isolation means that it must not be possible for an attacker, no matter how clever, to change the logic of the reference monitor or the contents of the security kernel database. Finally, the requirement for mathematical proof is formidable for something as complex as a general-purpose computer. A system that can provide such verification is referred to as a **trusted system.**

A final element illustrated in Figure 4.4 is an audit file. Important security events, such as detected security violations and authorized changes to the security kernel database, are stored in the audit file.

In an effort to meet its own needs and as a service to the public, the U.S. Department of Defense in 1981 established the Computer Security Center within the National Security Agency (NSA) with the goal of encouraging the widespread availability of trusted computer systems. This goal is realized through the center's Commercial Product Evaluation Program. In essence, the center attempts to evaluate commercially available products as meeting the security requirements just outlined. The center classifies evaluated products according to the range of security features that they provide. These evaluations are needed for Department of Defense procurements but are published and freely available. Hence, they can serve as guidance to commercial customers for the purchase of commercially available, off-the-shelf equipment.

4.7 Trojan Horse Defense

One way to secure against Trojan horse attacks is the use of a secure, trusted operating system. Figure 4.5 illustrates an example. In this case, a Trojan horse is used to get around the standard security mechanism used by most file management and operating systems: the access control list. In this example, a user named Doe interacts through a program with a data file containing the critically sensitive character string "CPE17O4TKS." User Doe has created the file with read/write permission provided only to programs executing on his own behalf: that is, only processes that are owned by Doe may access the file.

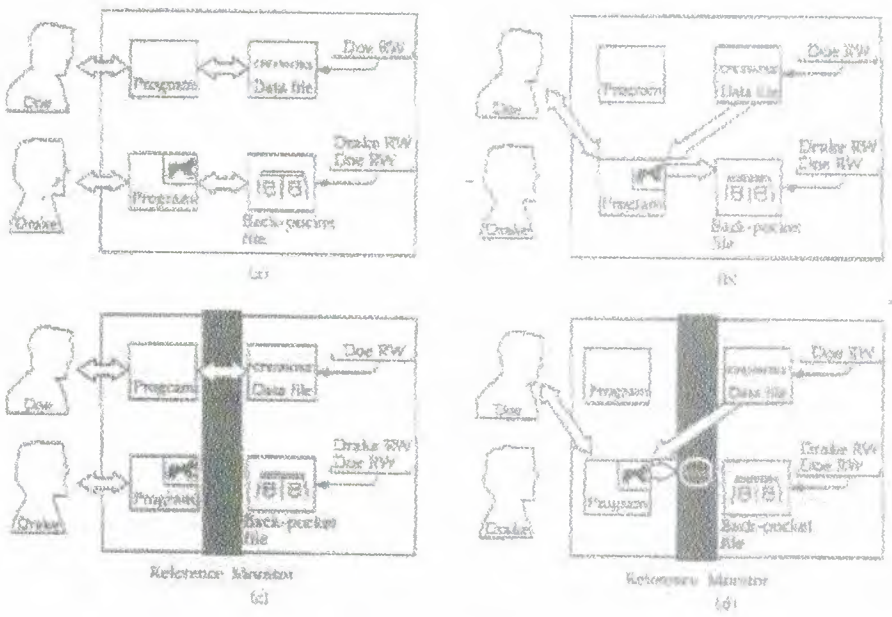

Figure 4.5 Trojan Horse and Secure Operating System.

The Trojan horse attack begins when a hostile user, named Drake, gains legitimate access to the system, and installs both a Trojan horse program and a private file to be used in the attack as a "back pocket." Drake gives read/write permission to himself for this file and gives Doe write-only permission (Figure 4.5a). Drake now induces Doe to invoke the Trojan horse program, perhaps by advertising it as a useful utility. When the program detects that it is being executed by Doe, it copies the sensitive character string from Doe's file and copies it into Drake's back-pocket file (Figure 4.5b). Both the read and write operations satisfy the constraints imposed by access control lists. Drake then has only to access his file at a later time to learn the value of the string.

Now consider the use of a secure operating system in this scenario (Figure 4.5c). Security levels are assigned to subjects at logon on the basis of criteria such as the terminal from which the computer is being accessed and the user involved, as identified by password/ID. In this example, there are two security levels, sensitive and public, ordered so that sensitive is higher than public. Processes owned by Doe and Doe's data file are assigned the security level sensitive. Drake's file and processes are restricted to public. If Doe invokes the Trojan horse program (Figure 4.5d), that program acquires Doe's security level. It is therefore able, under the simple security property, to

observe the sensitive character string. When the program attempts to store the string in a public file (the back-pocket file), however, the property is violated and the attempt is disallowed by the reference monitor. Thus, the attempt to write into the back-pocket file is denied even though the access control list permits it: The security policy takes precedence over the access control list mechanism.

CONCLUSIONS

Computer security threats were rare, and were basically concerned with insiders; these threats were well understood and dealt with using standard techniques. The key for building a secure network is to define what security means to your organization. Once that has been defined, everything that goes on with the network can be evaluated with respect to that policy. Projects and systems can then be broken down into their components, and it becomes much simpler to decide whether what is proposed will conflict with your security policies and practices.

Many people pay great amounts of lip service to security, but do not want to be bothered with it when it gets in their way. It's important to build systems and networks in such a way that the user is not constantly reminded of the security system around him. Users who find security policies and systems too restrictive will find ways around them. It's important to get their feedback to understand what can be improved, and it's important to let them know *why* what have been done has been, the sorts of risks that are deemed unacceptable, and what has been done to minimize the organization's exposure to them.

REFERENCES

[1] Patrick N. Smith , Client/Server Computing, SHL Systemhouse, Inc., United States of America, January 1994.

[2] Buzzard, James. The Client-Server Paradigm: Making Sense Out of the Claims, Data Based Advisor, August 1990.

[3] McGoveran, David. Evaluating an RDBMS for Client/Server Applications. Alternative Technologies, 1992.

[4] Adler, R. M. "Distributed Coordination Models for Client/Sever Computing." Computer 28, April 1995, pp. 14-22.

[5] Architecture. A Comparison of Two-Tier and Three-Tier Systems." *Information Systems Management Magazine",* Spring 1996.

[6] Brandel, M. Global Marine. "*Client/Server Journal",* 1995, pp. 25-27.

[7] Garner, R. Hughes Space communications. "*Client/Server Journal",* 1995, pp. 28-29.

[8] Schatz, W. Fannie Mae. "*Client/Server Journal",* 1995, pp.22-23.

[9] "Remapping the Computing Environment for the 90s: The Impact of Client/Server," Sybase Executive Summary, Winter 1993.

[10] Client/Server Computing with the AS/400. A Duke Communication International White Paper published as a supplement to News 3X/400, 1992.

[11] Lloyd Taylor, "comp. Client-ServerFAQ maintainer", "http://www.wp.com/Lloyd.Taylor". 1999.

[12] Schussel, George. "*Client/Server Past, Present, and Future*", "*http://www.dciexpo.com/geos*" , 1995.