NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

NETWORK SECURITY AND FIREWALLS

Graduation Project

COM- 400

Student: Khalil Horoub (20010691)

Supervisor: Dr. Murat Tezer

Nicosia – 2005

References

[1] S.M. Bellovin. Security Problems in the TCP/IP Protocol Suite. Computer Communication Review, Vol. 19, No. 2, pp. 32-48, April 1989.

[2] Strom, D. "The Packet Filter: A Basic Network Security Tool. September, 2000.

[3] The Use of Firewalls in an Academic Environment, T.Chown, J.Read, D.DeRoure, UKERNA, GD/JANET/TECH/002.

[4] B. Schneier. Applied Cryptography. John Wiley and Sons, New York, 1996.

[5] http://www.interhack.net/pubs/network-security/

[6] http://computer.howstuffworks.com/search.php?search=firewall&fr=ch

[7] http://www.altera.com/products/ip/dsp/encryption_decryption/ipm-index.jsp

[8] http://www.microsoft.com/technet/itsolutions/wssra/raguide/FirewallServices/Igfsbp_1.mspx

CONCLUSIONS

Security is a very difficult topic. Everyone has a different idea of what "security" is, and what levels of risk are acceptable. 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.

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.

CHAPTER ONE

INTRODUCTION TO 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.

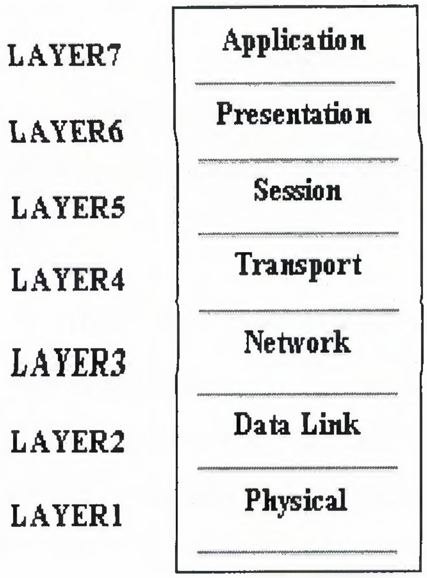

Figure 1.1 OSI Reference Model

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*.

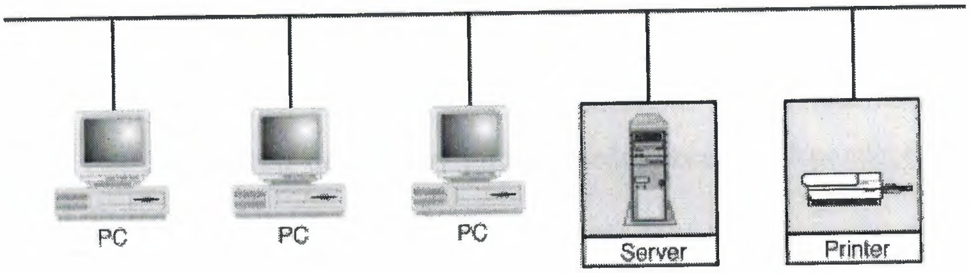

Figure 1.2 Typical bus topology.

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.

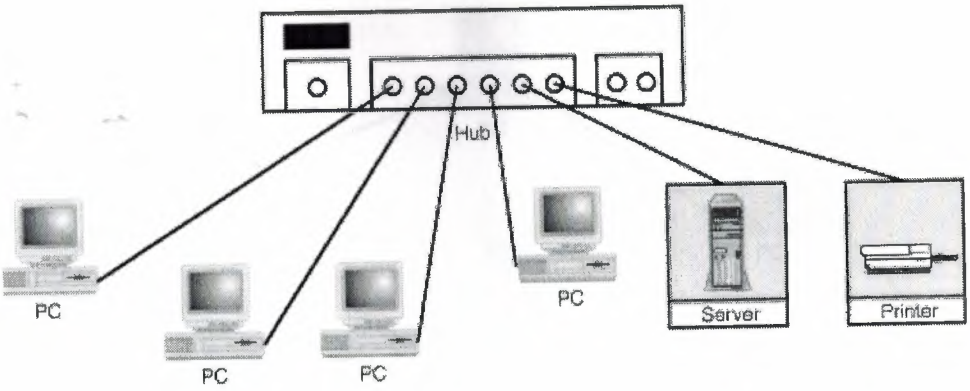

Figure 1.3 Star topology.

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."

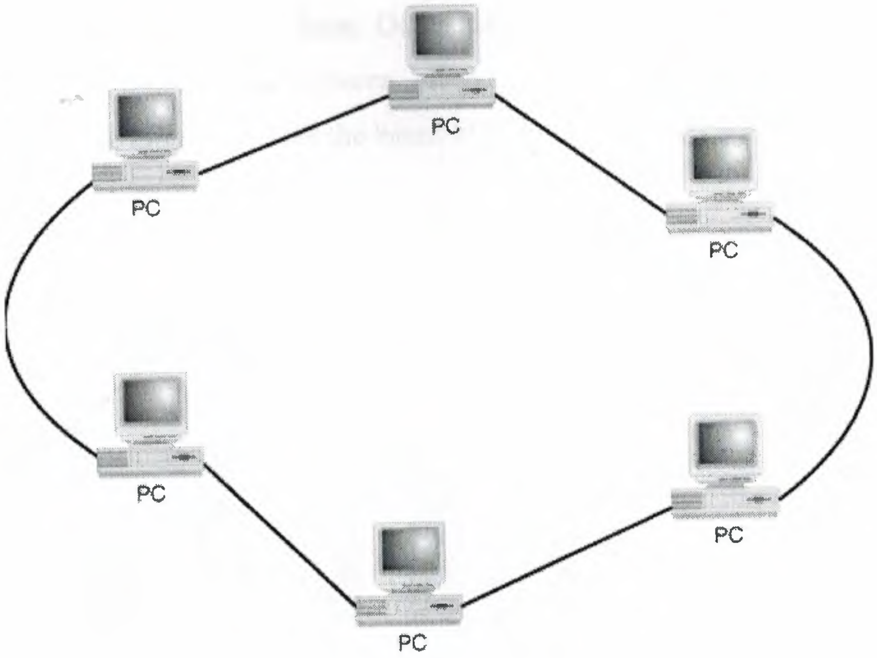

Figure 1.4. Peer-to-peer ring topology.

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.

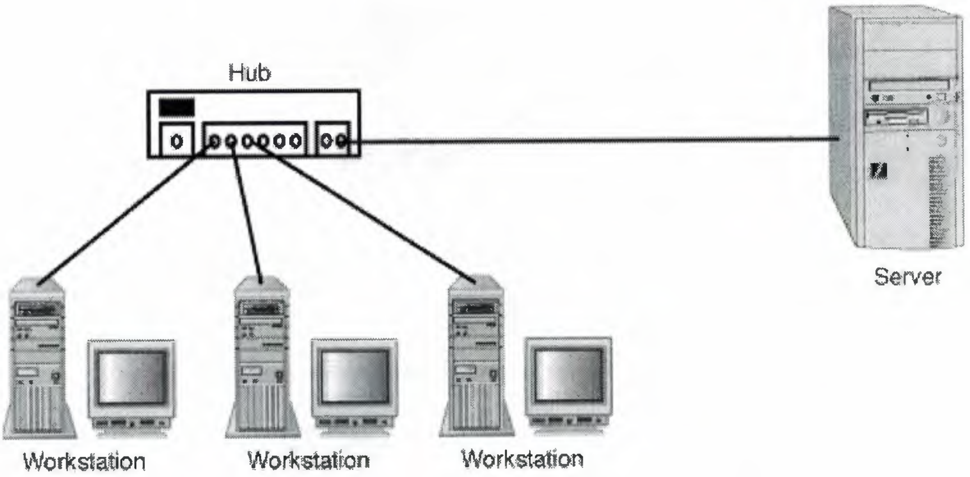

Figure 1.5. A basic diagram of a 10base-T network. Notice the hub, which is the device to which all systems initially connect.

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.

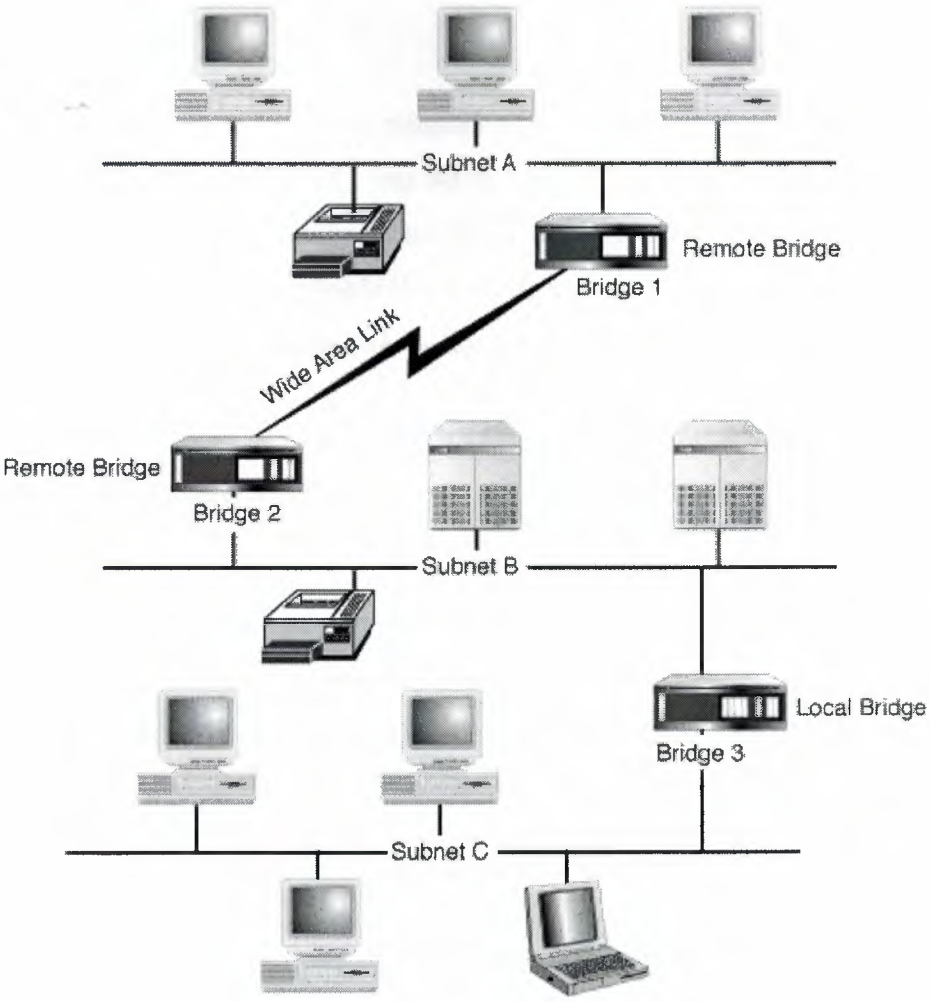

Figure 1.6 A sample network with local and remote bridges.

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.

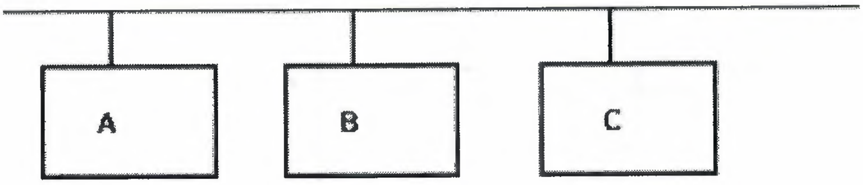

Figure 1.7 A Simple Local Area Network

I might 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.

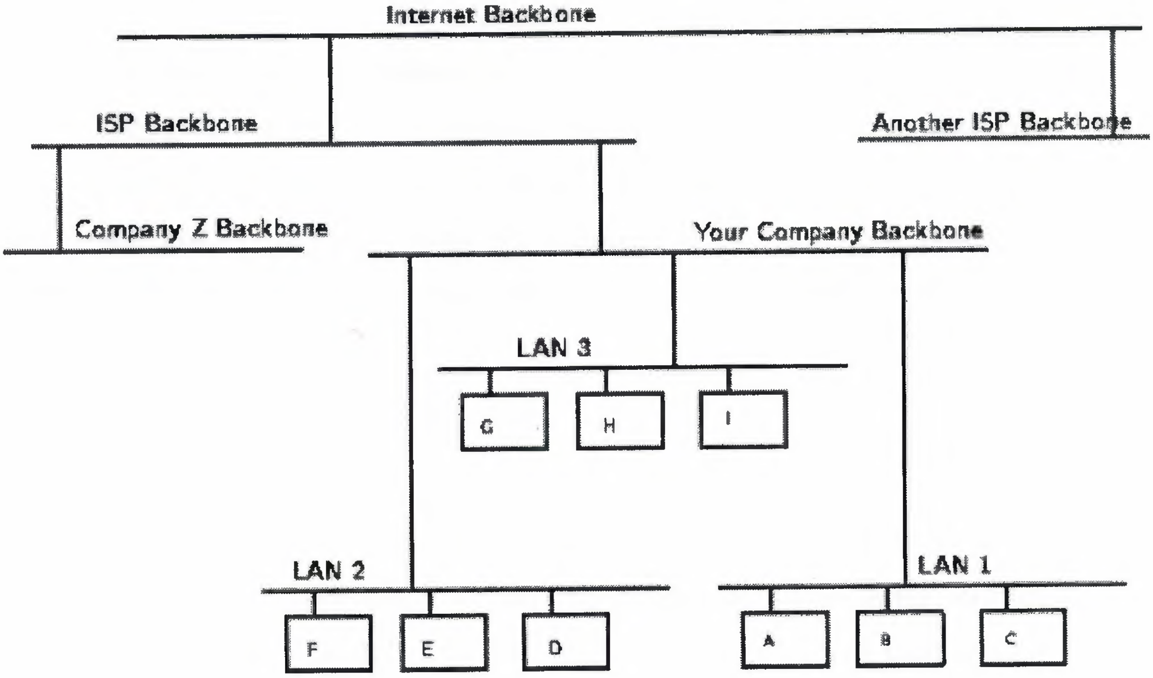

Figure 1.8 A Wider View of Internet-connected Network

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

NETWORK SECURITY

2.1 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.

2.2 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.

2.3 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

2.3.1 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.

2.3.1.1 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 *loop back* network (127.0.0.0).

2.3.1.2 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.

2.3.1.2.1 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.

2.3.1.2.2 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

2.3.2 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.

2.4 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.

2.4.1 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.

2.4.1.1 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.

2.4.1.2 Access 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.

2.4.1.3 Demilitarized Zone (DMZ)

The DMZ is a critical part of a firewall: it is a network that is neither part of the entrusted network, nor part of the trusted network. But, this is a network that connects the entrusted 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.

2.4.1.4 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.

2.4.1.5 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.

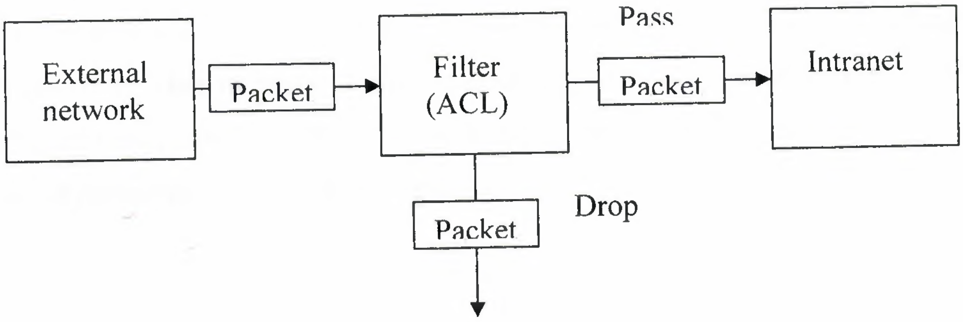

Figure 2.1 IP Filtering

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

2.5 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.

2.5.1 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.*

2.5.2 Virtual Private Networks (VPN)

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.

ACKNOWLEDGEMENT

First of all, I want to pay special regards to my parents who are enduring these all expenses and supporting me in all events. I am nothing without their prayers. They also encouraged me in crises. I shall never forget their sacrifices for my education so that I can enjoy my successful life as they are expecting. They may get peaceful life in Heaven. At the end I am again thankful to those all persons who helped me or even encouraged me to complete my project. My all efforts to complete this project might be fruitful.

More over, I feel proud to pay my special regards to my project adviser "Dr. Murat Tezer". He never disappointed me in any affair. He provided me too much information and did his best of efforts to make me able to complete my project, and I am less than the half without his help. I am really thankful to my teacher.

To the best of my knowledge, I want to honor those all persons who have supported me or helped me in my project. I also pay my special thanks to my all friends who have helped me in my project and gave me their precious time to complete my project.

ABSTRACT

Network security is a complicated subject, historically only tackled by well-trained and experienced experts. However, as more and more people become wired, an increasing number of people need to understand the basics of security in a networked world. This project explains the concepts needed in network security and how to understand risks and how to deal with them.

An introduction of networking is included, as well as an introduction to TCP/IP and internetworking . We go on to consider risk management, network threats, firewalls, and more special-purpose secure networking devices.

Firewall is a system designed to prevent unauthorized access to or from a private network. Firewalls can be implemented in both hardware and software, or a combination of both. Firewalls are frequently used to prevent unauthorized Internet users from accessing private networks connected to the Internet. We go on to consider also Basic Components of Firewal, Firewall Policy, Proxy Servers.

CHAPTER THREE

ELEMENTS OF SECURITY

3.1 Need for Network Security Policy

Before a network can be secured, a network security policy has to be established. A network security policy defines the organisation's expectations of proper computer and network use and the procedures to prevent and respond to security incidents. A network security policy is the foundation of security because it outlines what assets are worth protecting and what actions or inactions threaten the assets. The policy will weigh possible threats against the value of personal productivity and efficiency and identify the different corporate assets which need different levels of protection. Without a network security policy, a proper security framework cannot be established. Employees cannot refer to any established standards and security controls would be circumvented for the sake of increasing efficiency.

A network security policy should be communicated to everyone who uses the computer network, whether employee or contractor..

3.2 Risks of Network Connectivity

Before a network security policy can be established, a risk analysis has to be studied. Risk analysis is the process of identifying what you need to protect, what you need to protect it from, and how to protect it. It is the process of examining all of your risks, and ranking those risks by level of severity.

A good way of assessing the risks of network connectivity is to first evaluate the network to determine which assets are worth protecting and the extent to which these assets should be protected. In principle, the cost of protecting a particular asset should not be more than the asset itself. A detailed list of all assets, which include both tangible objects, such as servers and workstations, and intangible objects, such as software and data should be made.

Directories that hold confidential or mission-critical files must be identified. After identifying the assets, a determination of how much it cost to replace each asset must be made to prioritize the list of assets.

Once the assets requiring protection are identified, it is necessary to identify the threats to these assets. The threats can then be examined to determine what potential for loss exists

A thorough risk assessment will be the most valuable tool in shaping a network security policy. The risk assessment indicates both the most valuable and the most vulnerable assets. A security policy can then be established to focus on security measures that can identify these assets.

3.3 Components of a Network Security Policy

Although network security policies are subjective and can be very different for different organizations, there are certain issues that are relevant in most policies. This section explains some of the common components of a network security policy.

3.3.1 Cryptography

Cryptography is the science of using mathematics to encrypt and decrypt data. Cryptography enables you to store sensitive information or transmit it across insecure networks (like the Internet) so that it cannot be read by anyone except the intended recipient.

While cryptography is the science of securing data, *cryptanalysis* is the science of analyzing and breaking secure communication. Classical cryptanalysis involves an interesting combination of analytical reasoning, application of mathematical tools, pattern finding, patience, determination, and luck. Cryptanalysts are also called *attackers.*

3.3.1.1 Encryption and decryption

Data that can be read and understood without any special measures is called *plaintext* or *cleartext.* The method of disguising plaintext in such a way as to hide its substance is called *encryption.* Encrypting plaintext results in unreadable gibberish called *ciphertext.* 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 3.1* illustrates this process.

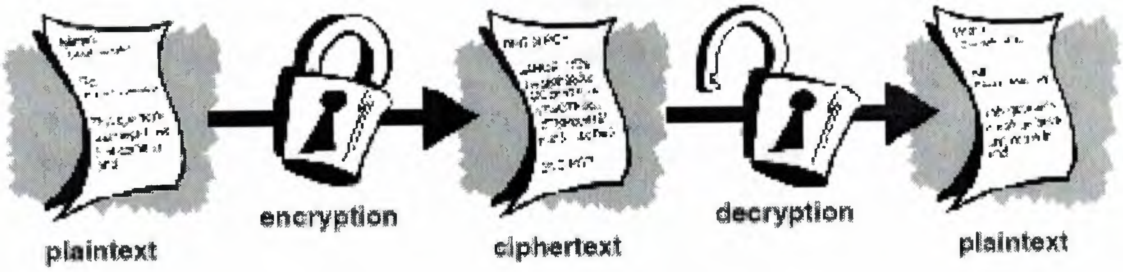

Figure 3.1 Encryption and decryption

3.3.1.2 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 ciphertext 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.

3.3.1.3 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.

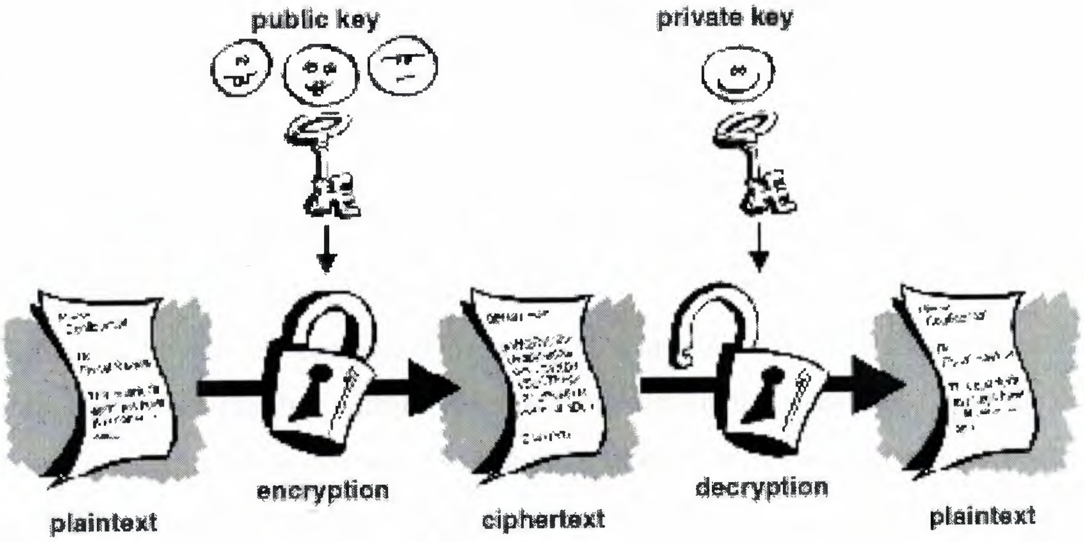

Figure 3.2 Public key encryption

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.

3.3.2 Choosing a Cryptographic Algorithm

A cryptographic algorithm is a procedure that takes the plaintext data and transforms it into cipher text in a reversible way.

3.3.2.1 Features of Cryptographic Algorithms

Cryptography includes a broad range of techniques under its umbrella. Some of the characteristics that distinguish one technique from another follow.

3.3.2.1.1 Level of protection

Some encryption techniques provide a virtually unbreakable barrier to information theft; others just require a determined attacker with moderate resources to be broken. One way to compare techniques on this level is to estimate how much CPU time would be required on a machine of a given processing speed to iterate through all the possible keys to the encoded data. For example, "A 128-bit XYZ cryptographic key requires 14.5 months of CPU time on an Acme 24-processor server to be broken." But other issues can affect the level of effort required to break the encrypted data, and make it difficult to objectively compare the security of encryption techniques. For example, if the attacker is not familiar with the format of the data being transmitted, and the data isn't easily interpreted on its own, then it may be tough to tell if an attempt to decode the data has worked or not.

3.3.2.1.2 Sophistication and complexity

Encryption techniques are usually based upon the mathematical properties of numbers and digital information. The mathematical theories employed in creating encryption techniques vary in their complexity and sophistication; some require poring over many pages of mathematics and statistics journals to be fully understood, while others can be explained using basic concepts of algebra. The resources required to implement and to break a given encryption technique are usually a direct function of its complexity. All other issues being equal, a more complex encryption scheme is normally more expensive to implement, but is also more expensive to break. As an application developer, you'll typically need to trade off efficiency against security. If high throughput is a requirement for your application, you may be willing to use a less complex and less secure, cryptographic algorithm if it imposes significantly less overhead on your agents.

3.3.2.1.3 One-, two-, and many-way cryptography

Depending on the nature of your distributed application, there may be situations in which many parties need to be individually or mutually authenticated by one agent. A secure chat room or whiteboard system, for example, may require mutual authentication by all participating parties. Most authentication schemes directly support one- or two-way agent authentication. Few, if any, have any concept of multi-way authenticated communications. Instead, the developer must maintain a set of two-way-authenticated channels at each agent site.

One way to deal with multi-way authentication is to define a group of individuals as a group identity with a single key pair and set of certificates. Every person in this group needs to prove ownership by providing a digital signature verifying that they have access to the group's private key, and some kind of certification of their keys. The viability of this approach really lies in the policies of the certification authority being used. If they agree to certify groups, and have defined a policy for verifying membership in certain groups, then most cryptographic algorithms will support a key pair or certificate associated with a named group rather than a named individual.

3.3.2.1.4 Design issues

As we've already mentioned, modern cryptography involves the use of keys for data signing, encoding, and decoding. Some require the secure distribution of private keys between parties, while others allow the parties to use public keys that can be broadcast openly. In our `SecureAgent` example, we used an encoding scheme that required the use of secret keys.

Other design issues involve the low-level implementation details of the algorithm. If we're building an agent to be used in an applet context, then we'd like the implementation of an cryptographic algorithm to be done completely in Java without any native method calls, so that we don't have to distribute native libraries to every agent that will talk to our agent. Another issue that may seem obvious is the level of standardization of a cryptographic algorithm. If an algorithm is standardized, or at least widely used, then there is a better chance that we can use our security-enabled agents with other people's agents without modifying them.

3.3.2.1.5 Financial and legal issues

An issue of a different sort is the expense of using a cryptographic algorithm. Some of the more sophisticated techniques are patented, and require payment of license fees to be used. Others are simply too involved to be implemented by the average developer, and software implementations can be sold for high prices by those who do have the resources to develop them.

To give an idea of the kind of price tags being put on high-caliber encryption packages, one set of Java encryption classes was listed at a cost of $25,000 for a limited-use license. Depending on the expected usage of the package, the price tag climbs as high as $100,000.

There are also legal and political restrictions to worry about. The United States, for example, has specific restrictions on the types of encryption that can be implemented for export to foreign countries. Currently, there are two separate versions of the Netscape browser: one for international use, and one for use only within the United States. The latter includes a more sophisticated version of the RSA encryption technology, whose export is restricted by the U.S. federal government.

3.3.2.2 Available Algorithms

There are numerous cryptographic algorithms for data encryption, and a similar number of certification and authentication techniques. Several umbrella security protocols and standards have been developed that incorporate both encryption and authentication facilities. Among them are Netscape's Secure Socket Layer (at the time of this writing, SSL 3.0 is an IETF Internet Draft), the Pretty Good Privacy (PGP) package developed originally by Phil Zimmermann, and the Public Key Cryptography Standard (PKCS) from RSA Laboratories.

S-HTTP is a security protocol designed specifically around the HTTP protocol, which differentiates it from these other, more general protocols. The chief motivation behind the development of these packages is to make encryption and authentication technologies easily accessible to developers, and to provide a common protocol for security-enabled applications to interact freely.

3.3.2.2.1 Encryption techniques

Some common forms of public key encryption in use today are Diffie-Hellman, derived from the original paper describing public key systems; RSA, licensed by RSA Laboratories; and the Digital Signature Algorithm (DSA), developed by the National Institute of Standards and Technology (NIST). Diffie-Hellman uses an encryption algorithm based on factoring prime numbers, while DSA is based on an algorithm involving discrete logarithms. These two algorithms are believed by many cryptographers to be comparably hard to crack. RSA is based upon a combination of their own public key encryption scheme with other secret key algorithms, such as DES block ciphers.

3.3.2.2.2 Certificates and authentication techniques

As we've already seen, certification and authentication schemes are typically founded on an existing public key encryption technique. The RSA public key cryptography system, for example, can be used in combination with a hashing technique to implement an authentication scheme. One party "signs" a message by running it through a known hashing algorithm, and then encrypts the hashed message with their private key to generate a digital signature. The signature is then sent along with the original message to the other party.

The receiving party can then decrypt the signature using the sender's public key, and hash the clear message using the same hashing algorithm used to generate the signature. If the hashed message is equal to the decrypted signature, then the receiving party can assume that the sender is the person that owns the public key. In order to impersonate some other party, we would have to know their private key. Otherwise, we wouldn't be able to generate both a clear message and an encrypted, hashed version of the message that check out against each other.

All authentication systems use certificates, which minimally contain an identifier for a party and the party's public key. The most commonly used standard format for digital certificates is the ITU X.509 standard certificate format. SSL, PKCS, and S-HTTP all offer X.509-compliant certification methods within their protocols. RSA can also be used in conjunction with X.509 certificates. Many cryptographic systems, however, resort to nonstandard certificate formats based on their own binary formatting schemes, where a public key, an identifier, and several other parameters, such as expiration times and serial numbers, are serialized and encrypted before being transmitted.

In most cases, the power of a certificate to authenticate a party rests on the certifying authority (CA) that issues and vouches for the certificate. VeriSign, BBN, and even the United States Postal Service offer CA services. Certificate authorities usually provide several levels of certification and require various types of proof of identity. In some cases email verification is sufficient; other certificates require verification by a notary public before being issued. A certificate issued from a CA must be installed somehow on your host computer in order to be properly broadcast by your network applications when secure transmissions are attempted. The Netscape browser, for example, will use your personal certificate to establish secure HTTP connections when a remote HTTP server requests it.

3.3.2.3 General Security Protocols

Some of the more common general security protocols have mentioned in use today, Support for them in the Java environment is becoming more broadly available, but implementation of the cryptographic algorithms underlying these protocols is a complex task that typically requires the backing of a software development company.

If public-domain or shareware versions of these protocols do become available they will undoubtedly be few and far between.

3.3.2.3.1 Secure Socket Layer (SSL)

The Secure Socket Layer, originally put forth by Netscape and now an Internet Draft of the IETF, defines a general security protocol for private, authenticated communications. SSL is a protocol that layers on top of a network transport protocol (such as TCP/IP), but below application protocols (such as HTTP) or security APIs (such as the Java Security API). The protocol allows agents to choose from a suite of encryption algorithms and authentication schemes, including DES ciphers and RSA cryptography.

3.3.2.3.2 Pretty Good Privacy (PGP)

PGP was originally developed by Phil Zimmermann as an effort to get effective privacy and authentication tools into the hands of the general technology community. There has been a shroud of controversy surrounding PGP since its public release by Zimmermann, involving United States export laws concerning cryptographic technology. We won't discuss the sordid details here, except to say that there are essentially two versions of PGP: one for use within the United States and Canada, and another for international use, which does not use certain implementations of RSA encryption algorithms.

In terms of the families of encryption algorithms that we discussed in an earlier section, PGP is a hybrid technique that uses public key methods to distribute private keys between agents. The reason for this is efficiency: public key methods are computationally very expensive, and can significantly reduce your data throughput if all data is transmitted with public key encryption.

PGP tries to circumvent this by using RSA public key encryption to transmit a pair of random private keys between the two parties securely. Once the agents have each other's private keys, they can proceed to encrypt their messages using faster, private key encryption. Messages are encrypted using a block cipher called IDEA. Note that, while SSL allows us to specify what kind of encryption we would like to use, the encryption scheme used by PGP is fixed.

Authentication in PGP can be implemented with digital signatures, using a method similar to that used by RSA for authentication. An MD5 hashing function and RSA encryption is used to generate the signature.

3.3.3 Authentication Methods

Your system has no security without authentication. Authentication means proving your identity. Authentication does not always have to be electronic. Locks, guards, and cameras can all provide authentication of some kind. None of these devices, however, are as constantly vigilant, carefully discriminating, or as fully reviewable as electronic methods are for protecting computer systems.

3.3.3.1 Post Name Check

The first and most simple type of authentication method is a *post name check.* The system checks where the user is coming from and uses that information to authenticate the user. In other words, the system has a secure list of trusted hosts, and anyone attempting to gain a connection from the trusted host can gain access, but users not from the trusted host are not allowed access. This method does have drawbacks, however, because it depends only on the physical security of one of the trusted hosts. If anyone can gain access to a trusted host, that user can then gain access to an individual computer in the system. In the early days of the Internet, this type of security was common.

3.3.3.2 Username Authentication

A slightly more secure method is *username authentication* in which the user merely types in his or her username; if the name is on the list, he or she is given access to the system.

An even more secure method, however, is *username and password authentication*, which allows the user to enter the username and password combination. This information is compared to a list that the computer has, and the user is then given access to the system if this information is the proper combination. You can use various twists on this arrangement to encrypt either part of that pair or both parts of the pair to make the system somewhat more secure. One example is the way in which UNIX stores passwords; in this approach, the username is stored in plain text, and the password is stored encrypted so that a user cannot steal the list and use it to gain access to the system. Encrypted passwords are very difficult to decrypt. Keep in mind that usernames and passwords need to be updated and changed every three months, because eventually they may be decrypted.

3.3.3.3 Kerberos

Another authentication method includes *Kerberos*. The name comes from the mythical name of the three-headed dog that guards the entrance to Hades. This method, primarily implemented under UNIX, is used to overcome problems with secure transmissions. It allows the user to be authenticated locally-that is, on the workstation-but to use network resources.

In the Kerberos system, the user puts in his or her username and password, and then the workstation itself authenticates the user. The workstation then requests from the Kerberos server a secret ticket for the user. This ticket is then used as a credential for any network resources. It is unique to the user for a specific time and situation. Transmitting this ticket is possible when the user wants to access certain resources that are protected. It is very secure because the user never transmits the username and password. Any eavesdroppers cannot steal the username and password, but instead get only an unusable ticket.

3.3.3.4 Smartcards

Smartcards, smartkeys, and what is known as a *challenge-and-response system* are protection methods similar to Kerberos. These systems create one-time usernames and passwords, which are the most secure. Challenge-and-response systems conduct all authentications on the local computer, avoiding transmission of passwords. Like kerberos, challenge-and-response systems create one-time passwords, but unlike kerberos, they do not require a special server.

3.3.4 Physical Security

Network security interacts with physical security because the size or shape of the network "machine" or entity can span a building, campus, country or the world due to interconnections and trust relationships. Without physical security, the other issues of network security like confidentiality, availability and integrity will be greatly threatened. The physical security section states how facilities and hardware should be protected. This section will also define which employees should be granted access to restricted areas such as server rooms and wiring closets.

3.3.5 Network Security

The network security section states how assets stored on the network will be protected. This section might include security measures regarding access controls, firewalls, network auditing, remote access, directory services, Internet services, and file system directory structures.

3.3.6 Access Control

Access control determines who has access to what. There must be a proper procedure to ensure that only the right people have access to the right information or services. Good access control includes managing remote access and enabling administrators to be efficient in their work. It should not be so complex that it becomes easy to commit errors.

3.3.7 Software Security

The software security section explains how the organisation will use commercial and non-commercial software on servers, workstations, and the network. This section might also identify who is allowed to purchase and install software and the security measures for downloading software from the Internet.

3.3.8 Auditing and Review

Once a security policy has been implemented, it must be checked to ensure that all components and employees are in compliance. Without sufficient auditing, an organization may have no legal recourse if there is a security breach. Auditing can also identify problems before they turn into security breaches. The policies must also be reviewed regularly to ensure that they are still relevant.

CHAPTER FOUR

FIREWALLS

4.1 Firewalls Overview

Firewalls are a very effective type of network security. This section briefly describes what Internet firewalls can do for your overall site security. describes the various types of firewalls in use today.

In building construction, a firewall is designed to keep a fire from spreading from one part of the building to another. In theory, an Internet firewall serves a similar purpose: it prevents the dangers of the Internet from spreading to your internal network. In practice, an Internet firewall is more like a moat of a medieval castle than a firewall in a modern building. It serves multiple purposes:

- It restricts people to entering at a carefully controlled point.
- It prevents attackers from getting close to your other defenses.
- It restricts people to leaving at a carefully controlled point.

An Internet firewall is most often installed at the point where your protected internal network connects to the Internet, as shown in figure 4.1.

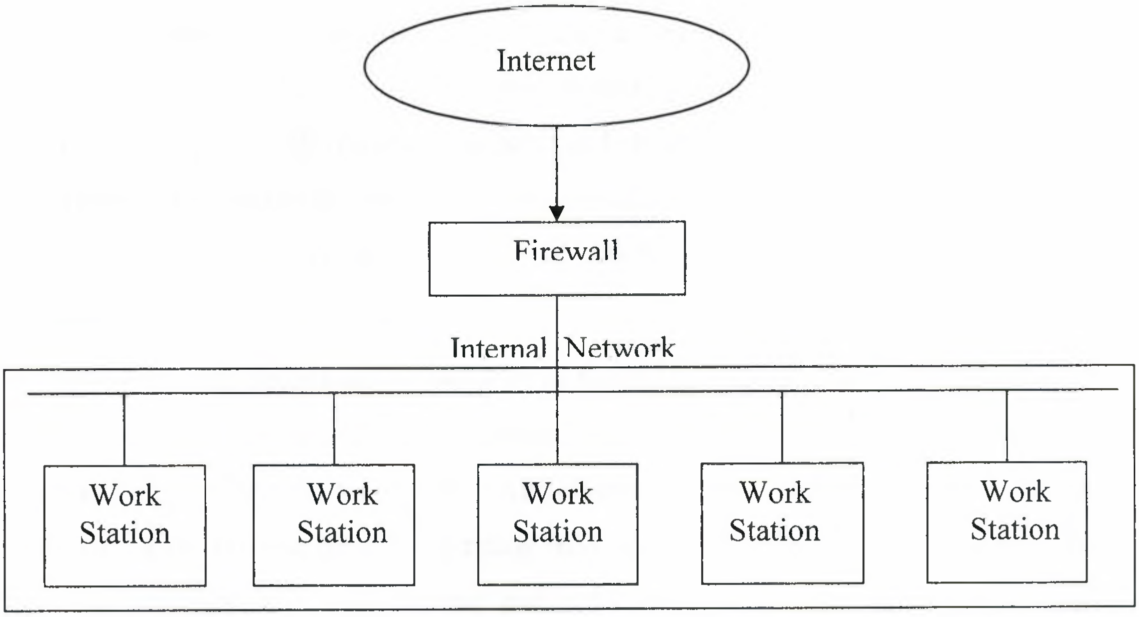

Figure 4.1 A firewall usually separates an internal network from the Internet

All traffic coming from the Internet or going out from your internal network passes through the firewall. Because it does, the firewall has the opportunity to make sure that this traffic is acceptable.

What does ”acceptable” mean to the firewall? It means that whatever is being done - email, file transfers, remote logins, or any kinds of specific interactions between specific systems - conforms to the security policy of the site. Security policies are different for every site; some are highly restrictive and others fairly open.

Logically, a firewall is a separator, a restricter, an analyzer. The physical implementation of the firewall varies from site to site. Most often, a firewall is a set of hardware components - a router, a host computer, or some combination of routers, computers, and networks with appropriate software. There are various ways to configure this equipment; the configuration will depend upon a site's particular security policy, budget, and overall operations.

A firewall is very rarely a single physical object, although some of the newest commercial products attempt to put everything into the same box. Usually, a firewall has multiple parts, and some of these parts may do other tasks besides function as part of the firewall. Your Internet connection is almost always part of your firewall. Even if you have a firewall in a box, it isn't going to be neatly separable from the rest of your site; it's not something you can just drop in.

We've compared a firewall to the moat of a medieval castle, and like a moat, a firewall is not invulnerable. It doesn't protect against people who are already inside; it works best if coupled with internal defenses; and, even if you stock it with alligators, people sometimes manage to swim across. A firewall is also not without its drawbacks; building one requires significant expense and effort, and the restrictions it places on insiders can be a major annoyance.

Given the limitations and drawbacks of firewalls, why would anybody bother to install one? Because a firewall is the most effective way to connect a network to the Internet and still protect that network. The Internet presents marvelous opportunities. Millions of people are out there exchanging information. The benefits are obvious: the chances for publicity, customer service, and information gathering. The popularity of the information superhighway is increasing everybody's desire to get out there. The risks should also be obvious: any time you get millions of people together, you get crime; it's true in a city, and it's true on the Internet. Any superhighway is fun only while you're in a car. If you have to live or work by the highway, it's loud, smelly, and dangerous.

Firewalls offer significant benefits, but they can't solve every security problem. The following sections briefly summarize what firewalls can and cannot do to protect your systems and your data.

4.2 What Can A Firewall Protect Against?

Some firewalls permit only email traffic through them, thereby protecting the network against any attacks other than attacks against the email service. Other firewalls provide less strict protections, and block services that are known to be problems.

Generally, firewalls are configured to protect against unauthenticated

Interactive logins from the “outside” world. This, more than anything, helps prevent vandals from logging into machines on your network. More elaborate firewalls block traffic from the outside to the inside, but permit users on the inside to communicate freely with the outside. The firewall can protect you against any type of network-borne attack if you unplug it.

Firewalls are also important since they can provide a single ``choke point'' where security and audit can be imposed. Unlike in a situation where a computer system is being attacked by someone dialing in with a modem, the firewall can act as an effective ``phone tap'' and tracing tool. Firewalls provide an important logging and auditing function; often they provide summaries to the administrator about what kinds and amount of traffic passed through it, how many attempts there were to break into it, etc.

This is an important point: providing this “choke point” can serve the same purpose on your network as a guarded gate can for your site's physical premises. That means anytime you have a change in “zones” or levels of sensitivity, such a checkpoint is appropriate. A company rarely has only an outside gate and any receptionist or security staff to check badges on the way in. If there are layers of security on your site, it's reasonable to expect layers of security on your network.

4.3 What Can't A Firewall Protect Against?

Firewalls can't protect against attacks that don't go through the firewall. Many corporations that connect to the Internet are very concerned about proprietary data leaking out of the company through that route. Unfortunately for those concerned, a magnetic tape can just as effectively be used to export data. Many organizations that are terrified (at a management level) of Internet connections have no coherent policy about how dial-in access via modems should be protected. It's silly to build a 6-foot thick steel door when you live in a wooden house, but there are a lot of organizations out there buying expensive firewalls and neglecting the numerous other back-doors into their network. For a firewall to work, it must be a part of a consistent overall organizational security architecture. Firewall policies must be realistic and reflect the level of security in the entire network. For example, a site with top secret or classified data doesn't need a firewall at all: they shouldn't be hooking up to the Internet in the first place, or the systems with the really secret data should be isolated from the rest of the corporate network.

Another thing a firewall can't really protect you against is traitors or idiots inside your network. While an industrial spy might export information through your firewall, he's just as likely to export it through a telephone, FAX machine, or floppy disk. Floppy disks are a far more likely means for information to leak from your organization than a firewall! Firewalls also cannot protect you against stupidity. Users who reveal sensitive information over the telephone are good targets for social engineering; an attacker may be able to break into your network by completely bypassing your firewall, if he can find a “helpful” employee inside who can be fooled into giving access to a modem pool. Before deciding this isn't a problem in your organization, ask yourself how much trouble a contractor has getting logged into the network or how much difficulty a user who forgot his password has getting it reset. If the people on the help desk believe that every call is internal, you have a problem.

Lastly, firewalls can't protect against tunneling over most application protocols to poorly written clients. There are no magic bullets and a firewall is not an excuse to not implement software controls on internal networks or ignore host security on servers. Tunneling ``bad'' things over HTTP, SMTP, and other protocols is quite simple and trivially demonstrated. Security isn’t “fire and forget”.

4.4 What Are Some Of The Basic Design Decisions In A Firewall?

There are a number of basic design issues that should be addressed by the lucky person who has been tasked with the responsibility of designing, specifying, and implementing or overseeing the installation of a firewall.

The first and most important decision reflects the policy of how your company or organization wants to operate the system: is the firewall in place explicitly to deny all services except those critical to the mission of connecting to the Net, or is the firewall in place to provide a metered and audited method of "queuing" access in a non-threatening manner? There are degrees of paranoia between these positions; the final stance of your firewall might be more the result of a political than an engineering decision.

The second is: what level of monitoring, redundancy, and control do you want? Having established the acceptable risk level by resolving the first issue, you can form a checklist of what should be monitored, permitted, and denied. In other words, you start by figuring

Out your overall objectives, and then combine a needs analysis with a risk assessment, and sort the almost always conflicting requirements out into a laundry list that specifies what you plan to implement.

The third issue is financial. We can't address this one here in anything but vague terms, but it's important to try to quantify any proposed solutions in terms of how much it will cost either to buy or to implement. For example, a complete firewall product may cost between $100,000 at the high end, and free at the low end. The free option, of doing some fancy configuring on a Cisco or similar router will cost nothing but staff time and a few cups of coffee. Implementing a high end firewall from scratch might cost several man-months, which may equate to $30,000 worth of staff salary and benefits. The systems management overhead is also a consideration. Building a home-brew is fine, but it's important to build it so that it doesn't require constant (and expensive) attention. It's important, in other words, to evaluate firewalls not only in terms of what they cost now, but continuing costs such as support.

On the technical side, there are a couple of decisions to make, based on the fact that for all practical purposes what we are talking about is a static traffic routing service placed between the network service provider's router and your internal network. The traffic routing service may be implemented at an IP level via something like screening rules in a router, or at an application level via proxy gateways and services.

The decision to make is whether to place an exposed stripped-down machine on the outside network to run proxy services for telnet, FTP, news, etc., or whether to set up a screening router as a filter, permitting communication with one or more internal machines. There are pluses and minuses to both approaches, with the proxy machine providing a greater level of audit and potentially security in return for increased cost in configuration and a decrease in the level of service that may be provided (since a proxy needs to be developed for each desired service). The old trade-off between ease-of-use and security comes back to haunt us with a vengeance.

4.5 Basic Components of Firewall

We have said that the firewall provides basic access control services for sites and corporate intranets. In accordance with specific security policy. The firewall intercepts data traffic and permits only the authorized and legitimate traffic to pass through. The access control services can be provided either at the network or transport layers using packet filtering, or at the higher using application gateways. In this section we overview the basic components a firewall typically consists of: a firewall policy, packet filters and application gateways.

4.5.1 Firewall Policy

There are two levels of policies that directly influence the design, installation and use of firewall system:

- The higher-level policy, the service access policy, defines the TCP/IP protocols and services that should be allowed or denied from the protected network, how these services should be used, and how exceptions to this policy are handling.
- The lower-level policy, that firewall design policy, describes how the firewall actually goes about restricting access and filtering the TCP/IP protocols and services according to the service access policy

Before we further address the two level of policy, we want to note that a firewall policy should always be as flexible as possible. This need for flexibility mainly due to the fact the internet itself is in flux, and that an organization's needs may change over time as the internet offers new services, methods and possibilities for doing business. New TCP/IP protocols and services are emerging on the internet, which offer more benefits to organization using the internet, but sometimes also result in new security concerns. Consequently, a firewall policy must be able to reflect and adequately address these concerns.

4.5.1.1 Service access policy

In short a network security policy (NSP) is a document that describes an organization's network security concerns and specifies the way network security should be achieved in that organization's environment. Parts of the NSP must include a service access policy that defines the TCP/IP protocols and services that should be accessible for internal and external use. As such, the service access policy extends the overall organization policy regarding the protection of information resources.

A firewall can implement a number of service access policies. Generally, a service access policy is focused more on keeping outsiders out than trying to police insiders. For example, a typical policy is to allow no inbound access to an intranet, but to allow full outbound access to the internet. Another typical policy would be to allow some inbound access from the internet, but perhaps only to selected systems, such as information servers or e-mail gateways. Also firewalls sometimes implement service access policy that allow access from the internet to selected internet systems, but this access would be granted only if necessary and only if it is combined with strong user authentication.

For a firewall to be successful, its service access policy must be realistic and reflect the level of security required for intranet. For example, a site with top secret and classified dada does not need a firewall at all. They should not be hooked up to the internet in the first place, or the systems with really secret data should be isolated from the rest of the intranet. A realistic service access policy in one that provides a balance between protecting intranet resources from the risks, while still protecting users access to external resources, such as internet.

The challenge is to find an appropriate balance between the accessibility and security of intranet resources, and this balance must be reflected in the service access policy of the corresponding firewall configuration.

4.5.1.2 Firewall design policy

The service access policy must be refined in a firewall design policy that is unique to a firewall configuration. The firewall design policy specifies the rules used by the firewall to implement the service access policy.

Formulating a firewall design policy as a difficult task, since one can not design it in a vacuum isolated from understanding issues such as firewall capabilities and limitations, as well as threats and vulnerabilities associated with the TCP/IP protocol and service. A key decision in the firewall design policy is the stance of the firewall design. The stance reflects the attitude of the firewall designers. It is determined by the cost of failure of the firewall and the designers' estimate of that likelihood. Obviously, it is also based on the designers' options of their own abilities. In general, a firewall may implement one of the following two stances:

- Permit any service unless it is expressly denied.
- Deny any service unless it is expressly permitted.

A firewall that implements the first stance allows TCP/IP protocols and services by default, with the exception of those that the service access policy identifies as disallowed. In other words, anything that is not expressly prohibited is permitted by default. From security point of view, this stance is less desirable, since it offers more avenues for circum \venting and getting around the firewall.

A firewall that implements the second stance denies TCP/IP protocols and services by default, and passes only those that are identified as allowed. Obviously this stance better fits the traditional access control model that is usually used in information security: anything that is not expressly permitted is prohibited by default. For a security point of view, this stance is preferable. However, that is usually also more difficult to implement and may impact users more ion that certain TCP/IP protocols and services must be blocked or restricted heavily.

4.5.2 Packet Filter Firewall

A packet filter firewall analyzes network traffic at the transport protocol layer. Each IP network packet is examined to see if it matches one of a set of rules defining what data flows are allowed. The rules determine whether communication is allowed based upon the information contained within the Internet and transport layer headers and the direction that the packet is headed. Packet filters enable the administrator to permit or prohibit the transfer of data based on the following controls: the physical network interface that the packet arrives on; the source IP address the data is coming from; the destination IP address the data is going to; the type of transport layer; the transport layer source port, and the transport layer destination port.

The packet filter architecture performs an analysis for one or more network protocols using a very limited rule set. The packets coming into the trusted network are compared against defined rules composed from a limited rule set for one or more protocols such as IP, TCP, or ICMP. Packets are either accepted and passed to the network stack for delivery or are denied access. If a packet satisfies all of the packet filter rules, the packet either moves up the network stack for future processing or gets forwarded to the network host. The rule set is maintained in the TCP/IP kernel. The rule set is used since packet filters do not generally understand the application layer protocols used in the communication packets. The rule set contains an associated action that will be applied to any packets matching the criteria established in the rule set. The rule set contains a deny list and a permit list which are maintained in the kernel. A network packet must first pass a check of both the deny and permit lists if it is to be routed to its proper destination. The packet must not be expressly denied and it must be expressly permitted.

Command sets that allow the checking of the source and destination port numbers on the TCP and UDP transport layer protocols are typically implemented by packet filters. The check is to determine if an applicable permit or deny rule exists for that specific port and protocol combination. It is difficult for packet filters to apply any security policy checking to the ICMP protocol layer since ICMP does not utilize port numbers for its communication protocol. To effectively apply the security policy to ICMP, the packet filter must maintain state tables to ensure that an ICMP reply message was recently requested from an internal host.

Packet filters generally do not understand how to process state information in the high level protocols such as FTP because the packet filters are implemented in the network layer. An administrator can permit certain types of connections to be made to specific computers while prohibiting other types of connections to those computers by using a packet filter that includes the TCP/UDP port filtering capability. In the general algorithm for complete network packet inspection if no matching rule is found the network packet is dropped; if a matching rule that permits the communication is found then peer to peer communication is allowed, or if a matching rule is found that denies the communication the network packet is dropped.

Packet filtering is the least secure firewall technology because it does not inspect the network packet's application layer data and does not track the state of connections. Packet filtering allows access through the firewall with a minimal amount of scrutiny. If the checks of the rules succeed the network packet is allowed to be routed through the firewall as defined by the rules in the firewall's routing table. Packet filters often use a process called network address translation to readdress network packets of outgoing traffic. This readdressing process makes the outgoing traffic appear to have originated from a different host rather than the internal host. The network readdress translation hides the topology and addressing schemes of trusted networks from untrusted networks.

Packet filtering firewall technology has several advantages. It is the fastest firewall technology since it performs fewer evaluations and does less processing than other technologies. It is often implemented in hardware components such as IP routers. By prohibiting connections between specific Internet sources and internal computers, a single rule in packet filtering can help protect an entire network. Packet filters do not require client computers to be specifically configured since the packet filter does all the work. Packet filter firewalls can be used to shield internal IP addresses from external users when used in conjunction with network address translation.

While the packet filtering firewall technology is the fastest technology it does have several disadvantages. Packet filter firewalls are less secure than application level firewalls because the packet filtering firewalls do not understand application layer protocols. Packet filtering firewalls cannot restrict access to protocol subsets for even the most basic services such as the PUT and GET commands in FTP. Packet filters do not inspect the payload of the packet. Decisions are not made based on the contents of the packet. The packet filter may allow dangerous forms of permissible traffic to pass through the firewall. An e-mail attachment that contains a virus could pass through the firewall if SMTP/POP connections are allowed. Packet filters are stateless since they do not keep application level information or information about a session. Packet filters have limited abilities to manipulate information within a packet. Packet filters do not offer higher level features such as HTTP object caching, URL filtering and authentication since packet filters do not understand the protocols being used and cannot discern one from another. Packet filters are not able to restrict the information that is passed from internal computers to services on the firewall server. Therefore intruders can potentially access the services on the firewall server. Packet filters have little or no audit event generation and alerting mechanisms. It can be difficult to test accept and deny rules of packet filters because of the complexity of supporting most non-trivial network services.

4.5.3 Application Level Firewall

An application level firewall evaluates network packets for valid data at the application layer before allowing a connection. The firewall examines the data in all network packets at the application layer and maintains complete connection state and sequencing information. Other security items such as user password and service requests that appear in the application layer data can be validated by the firewall. Specialized application software and proxy services are included in most application layer firewalls. Proxy services manage traffic through a firewall for a specific service such as HTTP or FTP. Proxy services can provide increased access control, detailed checks for valid data, and generate audit records about the traffic they transfer because the proxy services are specific to the protocol that they are designed to forward.

An application level firewall analyzes the complete command set for a single protocol in application space. When an incoming network packet is received it moves up the hardened network stack until it reaches the highest protocol layer found in the packet. After the network stack finishes processing the packet, its data is passed from kernel space to application space then to the proxy server that is listening on a specific TCP or UDP port. Next the proxy service processes the data it has received. The data is compared to the acceptable command set rules, as well as to host and user permission rules. The proxy determines whether to accept or deny the packet based on the results of the rules comparison. Based on how it was configured, the proxy may also perform other functions such as URL filtering, data modification, authentication logging, and HTTP object caching. A proxy service consists of the proxy server, proxy client and protocol analysis modes of operation.

A proxy server and a proxy client are two components that are typically implemented as a single executable for each application proxy. A proxy server acts as the end server for all connection requests originated on a trusted network by a real client. Rather than allowing users to communicate directly with the other servers on the Internet, all communication between the internal users on the trusted network and the Internet passes through the proxy server. When the internal user wants to connect to an external service such as FTP or Telnet they send a request to the proxy server for the connection. The proxy server decides whether to permit or deny the request based on an evaluation of a set of rules that is managed for the individual network service. Proxy servers only allow those packets through that comply with the protocol definitions because the servers understand the protocol of the service they are evaluating. The proxy client is the component that talks to the server on the external network on behalf of the real client on the trusted network. The proxy server evaluates a real client's request for a service against the policy rules defined for that proxy and determines whether to approve the request. The proxy server forwards the request to the proxy client if the request is approved. The proxy client contacts the real server on the external network on behalf of the client. The proxy client relays requests from the proxy server to the real server and relays responses from the real server to the proxy server. Then the proxy server relays the requests and responses between the proxy client and the real client.

Proxy services never allow direct connection between the real client on the trusted network and the real server on the external network. Proxy services force all network packets to be examined and filtered for suitability. All communication between the real user and the real service are handled by the proxy service. The proxy service is transparent to the user on the trusted network and the real service on the external network.

Proxy services are implemented on the top of the firewall host's network stack and operate only in the application space of the operating system. Proxy services are slower than packet filtering because each packet in a session is subjected to an examination process. Each network packet must pass through the low-level protocols in the kernel before being passed up the stack to application space. Once in the application space the proxies perform a thorough inspection of the packet headers and packet data. After inspection and acceptance the packet must travel back down to the kernel, and then back down the stack for distribution. Additional checks can be performed by application level firewalls to ensure that a network packet has not been spoofed. Application level firewalls can often perform network address translation.

Application level firewall technology using proxy services has several advantages. Proxy services enforce high level protocols such as HTTP and FTP. Information about the communications passing through the firewall server is maintained by the proxy service. Proxy services can permit access to certain network services, while denying access to others. Packet data can be processed and manipulated by proxy services. Internal IP addresses are shielded from the external world because proxy services do not allow direct communications between external server and internal computers. Administrators are able to monitor attempts to violate the firewall's security policies using the audit records that proxy services can generate.

Although application level firewalls provide increased security over a packet filtering firewall there are some disadvantages to using an application level firewall. Application level firewalls are slower since inbound data is processed by the application and by its proxy. A new proxy usually must be written for each protocol that is to pass through the firewall. This can cause the number of available network services and their scalability to be limited. Proxy services are vulnerable to operating system and application level bugs. Most application level firewalls require extensive support from the operating system to run correctly. The firewalls need support from TCP/IP, Win32, Winsock, NDIS, and the standard C library. The security of the firewall server can be effected by problems in these operating system components.

Another reason to use proxy servers is for the increased performance of the network. Proxies can cache documents that are frequently accessed by clients inside the firewall. Document caching is storing the most often requested file in memory where multiple users can access the document, rather than having to go get the document every time it is requested. This will cut down on the amount of disk space used, which will allow faster travel through the network. Caching will be discussed in further detail later in the report.

Proxies are placed between a physical connection point to the Internet and the connection point to the local network. What this means is that all communication from the network to the Internet or other Intranets is routed through the proxy. The actual workstations do not have a direct physical connection to the Internet, and therefore it is not possible for them to communicate in a direct way. The proxy server delivers the requests from the local net to the Internet as if it were the original requester. The proxy is a special HTTP server that typically runs on a firewall machine. It waits for requests from inside the firewall, and then sends them to the remote server, gets the response and sends it back to the client. This process is illustrated in the Figure 4.2 below.

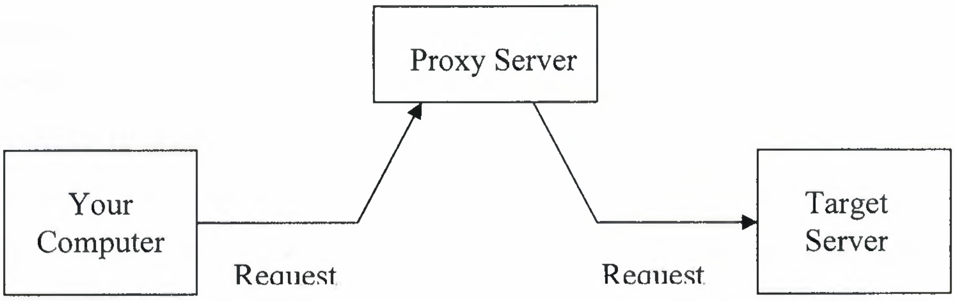

Figure 4.2 Proxy Server making a request

Proxy servers protect the network against hackers and those out to steal information from a company's internal network by disguising the client's IP address. The proxy does this by creating an address table where the client's true IP address is paired with a false IP address that is assigned to the proxy server. The philosophy behind the false IP address is that if an intruder were to capture the IP address during transmission of the message they would not actually gain access into the company's network. By using the fake address the intruder would only be able to access the proxy server. Below in table 4.2, is an example of an address table stored in the proxy server's memory.

Client IP Address	Proxy Assigned IP Address
3.3.3.55	192.168.21.21
3.3.3.57	192.168.21.22
3.3.3.47	192.168.21.23
Not Assigned	192.168.21.24
Not Assigned	192.168.21.25

Table 4.2 Address Table

The use of this table protects the IP addresses that are internal to the network from being "stolen" by users outside of the network. If by chance someone outside of the network were to gain access to these IP addresses the internal network would still be safe because none of the clients are physically connected to the Internet. The IP addresses the attacker would have would take them to some alternate site on the Internet. This is one of the main advantages to routing all Internet traffic through the proxy server.

The actual process a typical request would go through is comprised of many different steps that are all transparent to the user. The first step would be the actual request of an Internet site by the client. This client would travel through the subnet to the proxy server assigned to that subnet. Once the proxy receives the request it will take the client's IP address and store it into the address table. After storing the client's IP address the proxy will then substitute a fake IP address for the client's IP address and continue routing the request until it reaches its final destination. After the external sight processes the request the response is routed back to the proxy server. Now the proxy server will perform a series of data integrity checks to ensure that the information received from outside the network should be allowed into the internal network. After validating the data it searches the address table for the correct IP address and routes the information to the client. The entire process is transparent to the user and does not significantly slow down the processing and transmission speeds of the request.

4.5.3.1.1 Circuit-level Gateways

One step above standard packet filtering firewalls, but still considered part of the same architecture, are circuit level gateways, otherwise known as "stateful packet inspection" firewalls. In the circuit-level firewall, all connections are monitored and only those connections that are found to be valid are allowed to pass through the firewall.

This generally means that a client behind the firewall can initiate any type of session, but clients outside the firewall cannot see or connect to a machine protected by the firewall. Stateful inspections usually occur at the Network Layer, thus making it fast and preventing suspect packets from travelling up the protocol stack. Unlike static packet filtering, however, stateful inspection makes its decisions based on all the data in the packet (corresponding to all the levels of the OSI stack). Using this information, the firewall builds dynamic state tables.

It uses these tables to keep track of the connections that go through the firewall rather than allowing all packets that meet the rule set's requirements to pass, it allows only those packets which are part of a valid, established connection. Packet filtering firewalls are popular because they tend to be inexpensive, fast, and relatively easy to configure and maintain.

4.5.3.1.2 Application-Level Gateway

An application-level proxy server provides all the basic proxy features and also provides extensive packet analysis. When packets from the outside arrive at the gateway, they are examined and evaluated to determine if the security policy allows the packet to enter into the internal network. Not only does the server evaluate IP addresses, it also looks at the data in the packets to stop hackers from hiding information in the packets.

A typical application-level gateway can provide proxy services for applications and protocols like Telnet, FTP (file transfers), HTTP (Web services), and SMTP (e-mail). Note that a separate proxy must be installed for each application-level service (some vendors achieve security by simply not providing proxies for some services, so be careful in your evaluation). With proxies, security policies can be much more powerful and flexible because all of the information in packets can be used by administrators to write the rules that determine how packets are handled by the gateway. It is easy to audit just about everything that happens on the gateway. You can also strip computer names to hide internal systems, and you can evaluate the contents of packets for *appropriateness* and security.

4.5.2 Network Address Translation (NAT)

Firewalls using NAT and/or Port Address Translation (PAT) completely hide the network protected by the firewall by using many-to-one address translation. In most NAT implementations there is a single public IP address used for the entire network. All packets going outside the network have their internal IP addresses hidden for security, so any incoming packets are delivered to the network's public IP address. To handle ensuing port conflicts, PAT needs to be added to NAT.

NAT is the translation of an IP address used within one network to a different IP address known within another network. This is easier to understand by referring to the following diagram:

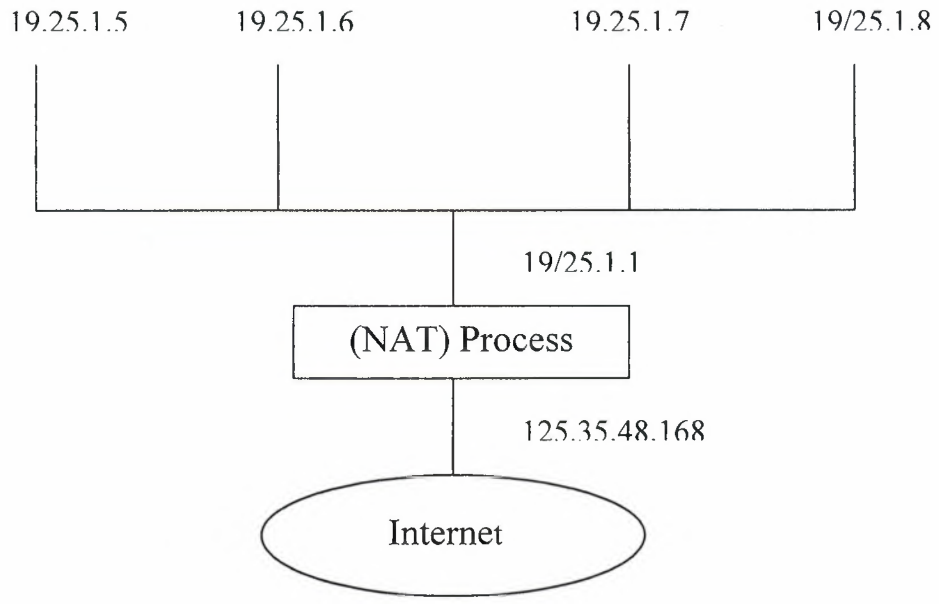

Figure 4.3 Network Address Translation

The NAT enabled router has an IP address of 10.25.1.1 for the inside network and an address of 125.35.48.168 for the outside network. Anytime a host on the inside network (10.25.1.0) makes a request to the outside network (the Internet in this instance) NAT will translate the 10.25.1.0 address to 125.35.48.168. From the inside looking out, the machines can access any host on the external network directly, while from the outside looking in it appears that all in and outbound traffic is originating from the single IP address on the router.

A disadvantage of NAT is that it can't properly pass protocols containing IP address information in the data portion of the packet.