NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

NETWORK SECURITY

Graduation Project
COM-400

Submitted By : **Baha Mahmoud (20002383)**

Supervisor : **Mr. Jamal Fathi**

Nicosia - 2005

ACKNOWLEDGEMENTS

First of all I would like to express sincere gratitude to my project advisor *and my brother "Dr.Jamal Fathi" for his patient and consistent support. Without his encouragement and direction, this work would not have been completed and I am really thankful to my doctor.*

More over I want to pay special regards to my family 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, I will never forget my father, my mother, my brother and my sisters. They may get peaceful life in Heaven.

Finally, the best of my acknowledges, I want to honor all my friends who have supported me or helped me in my life especially Ahmed Muslim. 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, especially Majed Abu-Zahrah, Netham Al-azzawi, Al-Katatbah(Abu-Rashed) and Murad Hassan.

ABSTRACT

The Internet has brought about many changes in the way organizations and individuals conduct business, and it would be difficult to operate effectively without the added efficiency and communications brought about by the Internet. At the same time, the Internet has brought about problems as the result of intruder attacks, both manual and automated, which can cost many organizations excessive amounts of money in damages and lost efficiency. Thus, organizations need to find methods for achieving their mission goals in using the Internet and at the same time keeping their Internet sites secure from attack.

Computer systems today are more powerful and more reliable than in the past; however they are also more difficult to manage. System administration is a complex task, and increasingly it requires that system administration personnel receive specialized training. In addition, the number of trained system administrators has not kept pace with the increased numbers of networked systems. One result of this is that organizations need to take extra steps to ensure that their systems are configured correctly and securely. And, they must do so in a cost-effective manner.

Active Network Security is comprised of a number of techniques that address this shortcoming. The goal is not only to reduce the number of successful abuses of a system, but also to give early warning of abuses in progress. Finally, the objective is to ensure that misuse of the system does not go unnoticed that, should all of the security mechanisms fail, a record exists to allow corrective action.

TABLE OF CONTENTS

ACKNOWLEDGEMENT i

ABSTRACT ii

TABLE OF CONTENTS iii

INTRODUCTION vii

1. NETWORK SECURITY 1

1.1 Overview 1

1.2 Security Risks 1

1.3 Network Threats 3

1.4 types and Sources of Network Threats 4

1.4.1 Denial of service 4

1.4.2 Unauthorized Access 4

1.4.3 Executing Commands Illicitly 5

1.4.4 Confidentiality Breaches 5

1.5 Firewalls 5

1.5.1 Bastion Host 6

1.5.2 Router 6

1.5.3 Access Control List 6

1.5.4 Demilitarized Zone 6

1.5.5 Proxy 7

1.5.6 IP Filtering 7

1.6 Types of Firewalls 9

1.6.1 Application Gateways 9

1.6.2 Packet Filtering 11

1.6.3 Hybrid System 12

1.7 Secure Network Devices 13

1.7.1 Secure Modems 13

1.7.1.1 Crypto-Capable Routers 14

1.7.1.2 Virtual Private Network 14

1.8 Summary 15

2. ELEMENTS OF SECURITY 16

2.1 Overview 16

2.2 Risks of Network Connectivity 16

2.3 Components of a Network Security Policy 17

2.3.1 Cryptography 17

2.3.2 Encryption and Decryption 17

2.4 How Does Cryptography Work ? 18

2.5 Public Key Cryptography 18

2.5.1 Authentication Methods 19

2.5.2 Post Name Check 20

2.5.3 User Name Authentication 20

2.5.4 Kerberos 21

2.5.5 Smartcards 21

2.5.6 Physical Security 21

2.5.7 Access Control 22

2.5.8 Software Security 22

2.6 Summary 22

3. FIREWALLS 23

3.1 Overview 23

3.2 Firewall Architectures 24

3.3 Benefit and Risks 26

3.4 What is a Firewall ? 31

3.5 What are the Typical Firewall Components ? 32

3.5.1 Typical Configuration 34

3.5.2 What about Notes and Domino ? 37

3.5.3 Packet Filtering 38

3.5.4 Circuit Level Proxy 38

3.5.5 Application Level Proxy 38

3.5.6 HTTP Proxy 38

3.5.7 Passthru 39

3.6 Summary 39

4. ACTIVATE NETWORK SECURITY 40

4.1 Overview 40

4.2 Active Security Mechanisms 40

4.3 The Limitation of Network security 41

4.3.1 Authentication 41

4.3.2 Cryptography 42

4.3.3 Access Control 43

4.3.4 Firewalls 43

4.4 What Do Static Methods Offer 43

4.5 The limitations of Static Security 44

4.5.1 Sources of Attack 45

4.5.2 Outline of an Attack 47

4.5.2.1 Exploring The Target 48

4.5.2.2 Vulnerability Identification 49

4.5.2.3 Penetration 50

4.5.2.4 Escalation 50

4.5.2.5 Embedding 50

4.5.2.7 Relay 51

4.5.2.6 Extraction 51

4.6 Typical Attack Techniques 52

4.7 Policy Issues for Active Security 55

4.7.1 What is Security Policy? 55

4.7.2 The Relationship between Active Security and Security Policy 56

4.8 Tools Supporting Active Security 57

4.8.1 Network Mappers 57

4.8.2 Network Security Scanners 58

4.8.3 System Integrity Checkers 59

4.8.4 Password Crackers 61

4.8.5 Sniffer Detection 62

4.8.6 Honeytrap Systems 64

4.9 Summary 65

5. INTERNET SECURITY 66

5.1 Overview 66

5.2 Basic Security Concepts 66

5.3 Why Care About Security? 68

5.4 Network Security Incidents 68

5.5 Sources of Incidents 69

5.6 Types of Incidents 69

5.7 Incident Trends 73

5.8 Intruders' Technical knowledge 73

5.9 Techniques to Exploit Vulnerabilities 74

5.10 Intruders' Use of Software Tools 75

5.11 Internet Vulnerabilities 77

5.11.1 Why the Internet Is Vulnerable 77

5.11.2 Types of Technical Vulnerabilities 79

5.12 Flaws in Software or protocol Designs 79

5.12.1 Weaknesses in How Protocols and Software Are Implemented 79

5.12.2 Weaknesses in System and Network Configuration 80

5.13 Security Policy, Procedures, and Practices 81

5.13.1 Security Policy 81

5.13.2 Security-Related Procedures 82

5.13.3 Security Practices 82

5.13.4 Security Technology 83

5.14 Operational Technology 83

5.15 Information Warfare 85

5.16 Summary 86

CONCLUSION 87

REFERENCES 88

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, five chapters and conclusion.

1. NETWORK SECURITY

1.1 Overview

So far the terminology has been restricted to encryption and decryption with the goal of privacy in mind. Network security is much broader, encompassing such things as authentication and data integrity.

- A network security service is a method to provide specific aspect of security.
- Breaking a network security service implies defeating the objective of the intended service.
- A passive adversary is an adversary who is capable only of reading information from an unsecured channel.
- An active adversary is an adversary who may also transmit, alter, or delete information on an unsecured channel.

1.2 Security Risks

Information security is concerned with three main areas:

- Confidentiality : information should be available only to those who rightfully have access to it
- Integrity : information should be modified only by those who are authorized to do so
- Availability : information should be accessible to those who need it when they need it

These concepts apply to home Internet users just as much as they would to any corporate or government network. You probably wouldn't let a stranger look through your important documents. In the same way, you may want to keep the tasks you perform on your computer confidential, whether it's tracking your investments or sending email messages to family and friends. Also, you should have some assurance that the information you enter into your computer remains intact and is available when you need it.

Some security risks arise from the possibility of intentional misuse of your computer by intruders via the Internet. Others are risks that you would face even if you weren't connected to the Internet (e.g. hard disk failures, theft, power outages). The bad news is that you probably cannot plan for every possible risk. The good news is that you can take some simple steps to reduce the chance that you'll be affected by the most common threats -- and some of those steps help with both the intentional and accidental risks you're likely to face. Before we get to what you can do to protect your computer or home network, let's take a closer look at some of these 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.

1.3 Network 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.

1.4 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.

1.4.1 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.

1.4.2 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.

1.4.3 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.

1.4.4 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.

1.5 Firewalls

As we've seen in our discussion of the Internet and similar networks, connecting an organization to the Internet provides a two-way flow of traffic. This is clearly undesirable in many organizations, as proprietary information is often displayed freely within a corporate intranet (that is, a TCP/IP network, modeled after the Internet that only works within the organization).

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.

1.5.5 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 <http://www.interhack.net/> 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.

1.5.6 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 as shown in figure 1.1. 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.

In order to provide some level of separation between an organization's intranet and the Internet, firewalls have been employed. A firewall is simply a group of components that collectively form a barrier between two networks.

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

1.5.1 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.

1.5.2 Router

A special purpose computer for connecting networks together. Routers also handle certain functions, such as routing , or managing the traffic on the networks they connect.

1.5.3 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.

1.5.4 Demilitarized Zone (DMZ)

The DMZ is a critical part of a firewall: it is a network that is neither part of the untrusted network, nor part of the trusted network. But, this is a network that connects the untrusted to the trusted. The importance of a DMZ is tremendous: someone who breaks into your

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.

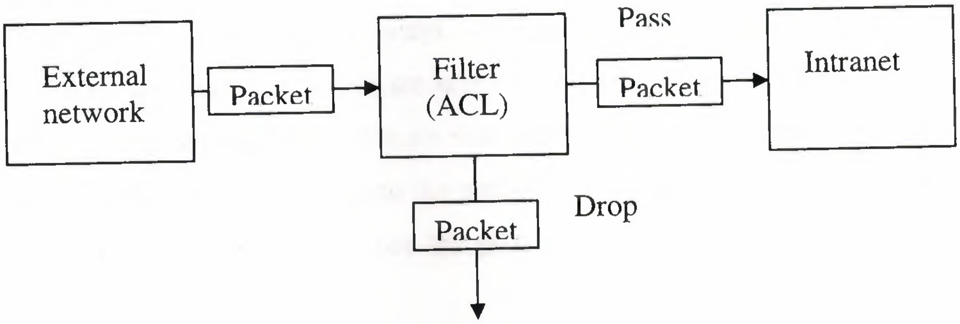


Figure1.1 IP Filtering

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.

1.6 Types of Firewalls

There are three basic types of firewalls, and we'll consider each of them.

1.6.1 Application Gateways

The first firewalls were application gateways, and are sometimes known as proxy gateways as described in figure 1.2. These are made up of bastion hosts that run special software to act as a proxy server. This software runs at the Application Layer of our old friend the ISO/OSI Reference Model, hence the name. Clients behind the firewall must be proxitized (that is, must know how to use the proxy, and be configured to do so) in order to use Internet services.

Traditionally, these have been the most secure, because they don't allow anything to pass by default, but need to have the programs written and turned on in order to begin passing traffic.

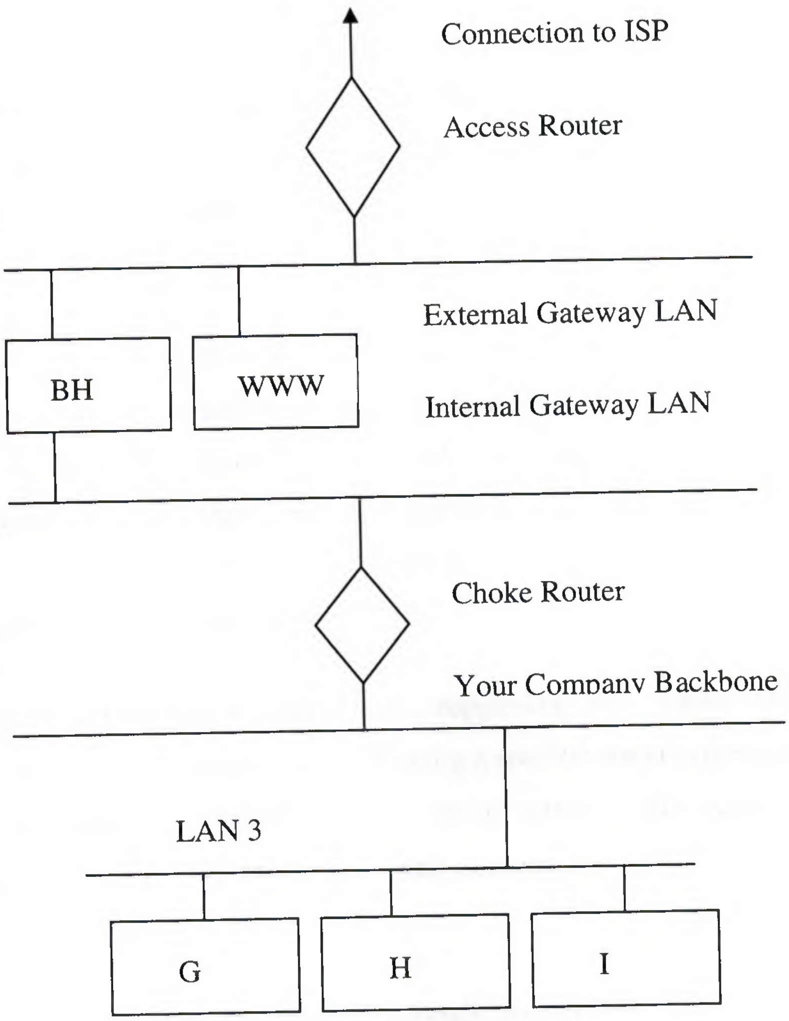


Figure1.2 A Sample Application Gateway

These are also typically the slowest, because more processes need to be started in order to have a request serviced.

1.6.2 Packet Filtering

Packet filtering is a technique whereby routers have ACLs (Access Control Lists) turned on. By default, a router will pass all traffic sent it, and will do so without any sort of restrictions. Employing ACLs is a method for enforcing your security policy with regard to what sorts of access you allow the outside world to have to your internal network, and vice versa.

There is less overhead in packet filtering than with an application gateway, because the feature of access control is performed at a lower ISO/OSI layer (typically, the transport or session layer). Due to the lower overhead and the fact that packet filtering is done with routers, which are specialized computers optimized for tasks related to networking, a packet filtering gateway is often much faster than its application layer cousins. Figure 6 shows a packet filtering gateway.

Because we're working at a lower level, supporting new applications either comes automatically, or is a simple matter of allowing a specific packet type to pass through the gateway. (Not that the possibility of something automatically makes it a good idea; opening things up this way might very well compromise your level of security below what your policy allows.)

There are problems with this method, though. Remember, TCP/IP has absolutely no means of guaranteeing that the source address is really what it claims to be. As a result, we have to use layers of packet filters in order to localize the traffic. We can't get all the way down to the actual host, but with two layers of packet filters, we can differentiate between a packet that came from the Internet and one that came from our internal network. We can identify which network the packet came from with certainty, but we can't get more specific than that.

1.6.3 Hybrid Systems

In an attempt to marry the security of the application layer gateways with the flexibility and speed of packet filtering, some vendors have created systems that use the principles of both. Figure 1.3 shows a sample packet filtering gateway.

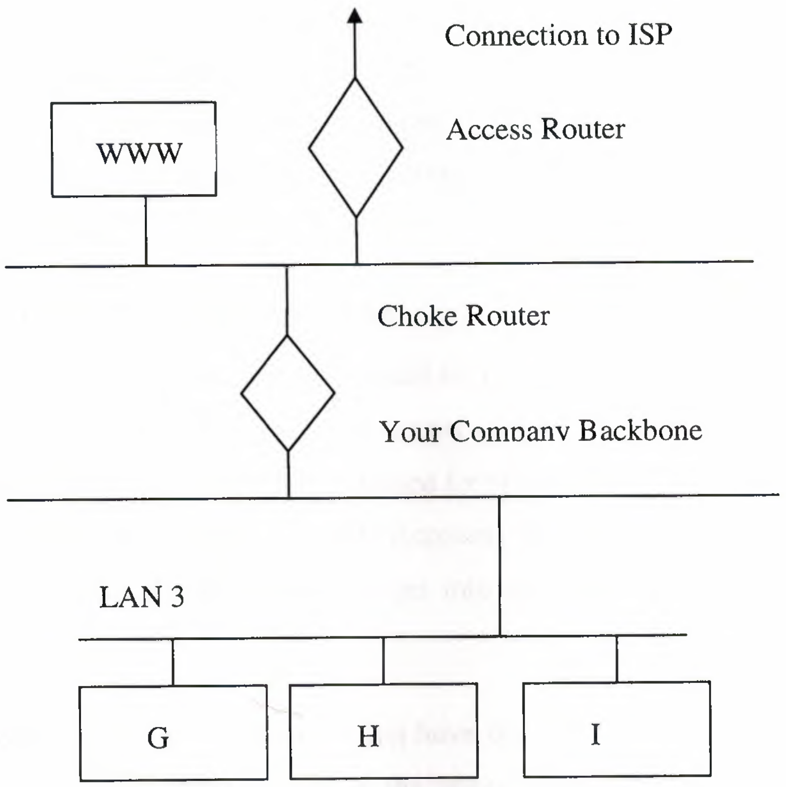


Figure1.3 A Sample Packet Filtering Gateway

In some of these systems, new connections must be authenticated and approved at the application layer. Once this has been done, the remainder of the connection is passed down to the session layer, where packet filters watch the connection to ensure that only packets that are part of an ongoing (already authenticated and approved) conversation are being passed.

Other possibilities include using both packet filtering and application layer proxies. The benefits here include providing a measure of protection against your machines that provide services to the Internet (such as a public web server), as well as provide the security of an application layer gateway to the internal network. Additionally, using this method, an attacker, in order to get to services on the internal network, will have to break through the access router, the bastion host, and the choke router.

1.7 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.

1.7.1 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.

1.7.1.1 Crypto-Capable Routers

A feature that is being built into some routers is the ability to session encryption between specified routers. Because traffic traveling across the Internet can be seen by people in the middle who have the resources (and time) to snoop around, these are advantageous for providing connectivity between two sites, such that there can be secure routes.

1.7.1.2 Virtual Private Networks

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.

1.8 Summary

In This chapter we discussed about network security ,the risks of network security, type and sources of network threats, firewalls and its types and secure network devices.

2. ELEMENTS OF SECURITY

2.1 Overview

Before a network can be secured, a network security policy has to be established. A network security policy defines the organization'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..

2.2 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.

2.3 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.

2.3.1 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.

2.3.2 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 cipher text to its original plaintext is called decryption. Figure 2.1 illustrates this process.

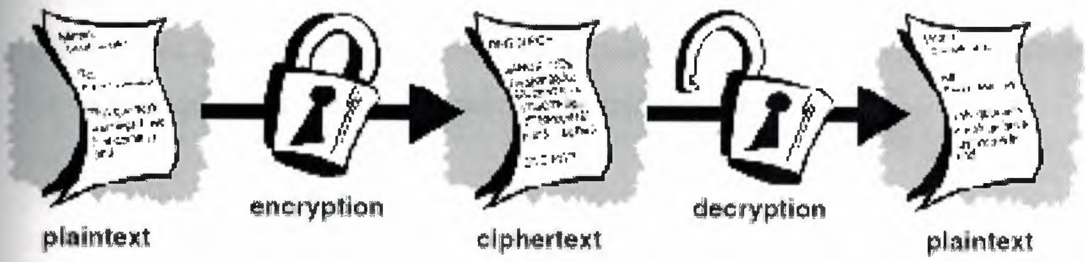


Figure2.1 Encryption and Decryption

2.4 How Does Cryptography Work?

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

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

2.5 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])

Figure 2.2 explain public key cryptography which 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.

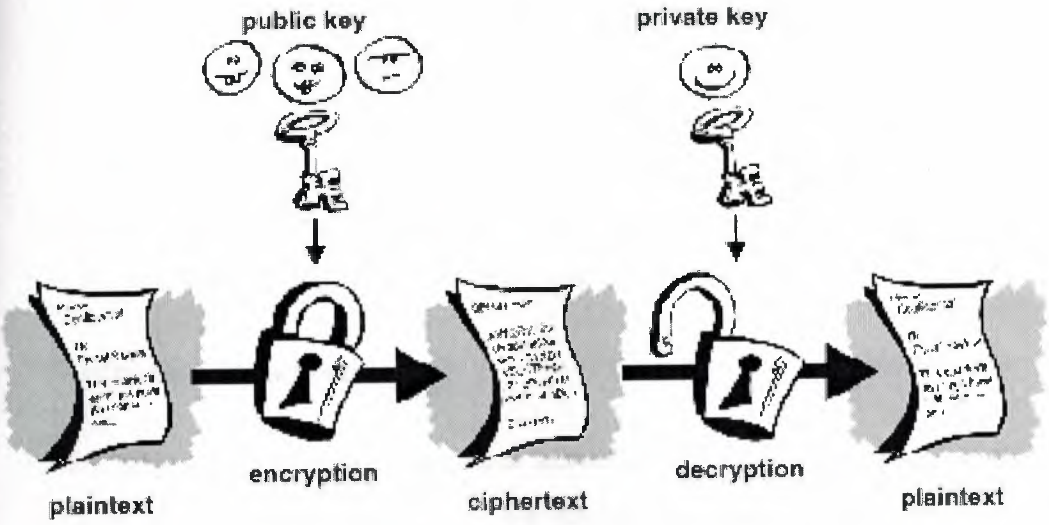


Figure2.2 Public Key Encryption

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

2.5.1 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.

2.5.2 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.

2.5.3 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.

2.5.4 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.

2.5.5 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.

2.5.6 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.

2.5.7 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.

2.5.8 Software Security

The software security section explains how the organization 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.

2.6 Summary

In this chapter we discussed the elements of security, risks of network connectivity and network security policy and the most common of a network security policy.

3.FIREWALLS

3.1 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.

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.

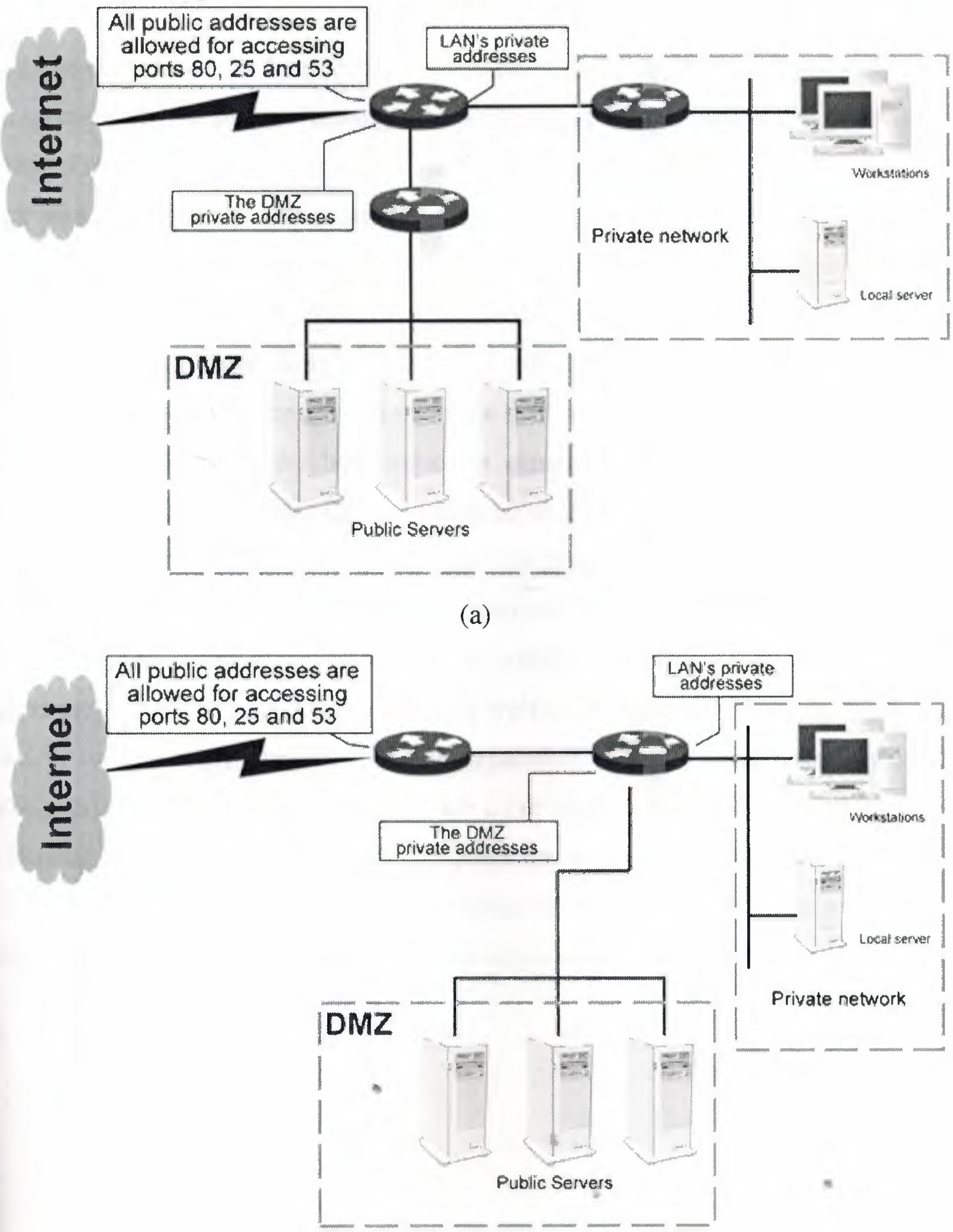
3.2 Firewall Architectures

Imagine a LAN as a building with its size in proportion to the computer network size and capacity. The building has its offices – workstations, store rooms and archive rooms servers, corridors that connect various building segments – routers, the guard hut the Demilitarized Zone (DMZ). When implementing a defensive system for building security, the designer must plan the positioning of firewalls in advance so that they will be able to block a fire and protect as much of the building structure as possible. It's obvious, that all walls of the building might be made of a firewall technology, but the costs involved would become magnified out of all proportion. Striking a happy medium is necessary. Therefore, when considering firewall deployment, the designer must well address the following question: "From where would a threat to my system most likely originate and for what reasons?" Once the places of potential origin of the fire have been determined, the designer can attempt to make a layout of firewalls. The similarities end there however. The designer of a building is allowed to be free from the fear that a disgruntled employee might set off a fire in the office using the furniture, whilst on the other hand the firewall designer will have to take into consideration such events.

Many users inside the protection of a firewall may believe that their systems are safe, since the firewall sits between the LAN and the public network. This is risky thinking; because firewalls are perimeter security only (even those being equipped with "true" firewall features) and once bypassed provide little or no security. A firewall based on a "better than nothing" philosophy runs the considerable risk that may provide a false sense of security. If you are considering implementing a "true" firewall, remember that a consistent security policy must be outlined in advance and this is not a concern of the elaboration methodology but of its essence. The security policy must determine how basic communication will take place at the firewall, where the firewall must sit and how to configure it. The security policy should also define if more than one firewall is required (or maybe, that a firewall would be of no use) and what should the connectivity scheme be. Once installed, a firewall system is an ongoing process that requires constant vigilance, maintenance, log reviewing and response to events. The inability to keep these requirements satisfied, and sometimes made worse by an inadequate or poor administration that would weaken any protection provided by even the best firewall, would result in it becoming nothing but a murmuring and flashing electronic box, yet adding the danger of providing the illusion of security that can further erode the private network itself. Firewalls are typically implemented using two approaches. The firewall literature is full of theories that categorize firewalls as hardware-based and software-based ones but there is nothing in such a classification that reasonably suggests a hierarchical point of view. I think instead, that a less debatable and apt classification will be that of using the notions of a dedicated and non-dedicated firewall hardware and system platform. Such an implementation approach may become an important factor in choosing a firewall solution, although the very decision must be taken directly by an experienced and knowledgeable system administrator or person installing the firewall. A must-have for any non-dedicated firewall application system is a proper installation of the operating system on which the firewall will be placed. A “proper installation” means that the operating system must be suitably “hardened” (i.e. configured for security) and especially for this reason, no service going beyond the necessary minimum may be run on the operating system. With dedicated firewall hardware and software platforms, it is very likely, that they are sold with their minimum protection (without useless overheads) built in by the manufacturer and ready to power on and configure. This does not imply however, that turnkey solutions are always better than non-dedicated own applications, since commercial products might not be free of manufacturer’s errors, and as such, more difficult to be debugged in respect to non-dedicated tools. So in this case, firewall management is also a critical issue because the firewall administrator must not only know how to manage a firewall, but also how to maintain and upgrade it for security. Another important consideration in implementing a firewall is a reduced capacity of key network nodes.

3.3 Benefits and risks

A firewall is primarily used to protect the boundary of an organization’s internal network whilst it is connected to other networks (e.g. to the Internet). A typical misconception is, already mentioned, to use perimeter routers for performing this role. At the very least, perimeter routers can be employed in two ways: either without packet filtering rules involved or by using an IP filtering router solution (most likely together with a dynamic NAT) selectively passing or blocking data packets based on port information or addresses acceptable by the security policy. Of course, a firewall must always be situated next to the router. Some practical solutions to this are illustrated in Figures 3.1(a) and (b) below.


Internet
All public addresses are allowed for accessing ports 80, 25 and 53
LAN's private addresses
The DMZ private addresses
Workstations
Private network
Local server
DMZ
Public Servers
Internet
All public addresses are allowed for accessing ports 80, 25 and 53
LAN's private addresses
The DMZ private addresses
Workstations
Local server
Private network
DMZ
Public Servers

(a)

(b)

Figure 3.1 Some Practical Solution

(a) Without Packet Filtering (b) IP Filtering Router

In these examples, a perimeter router controls traffic at the IP level. I think this device should be considered the first (but not only) line of defense protecting a private network. In implementing the packet filtering mechanism, it is a good idea to run this service on perimeter routers placed inside private networks (that separate two networks) primarily to block unwanted packets accessing other LANs. The criteria used in filtering rules for determining the disposition of packets (accept or reject) should be consistent with the specific security policy, not established at the discretion of the system administrator. In each of the figures there is an isolated area called DMZ that stands for Demilitarized Zone. A DMZ in the IT sense is an interface that enables the network designer to setup different rules of access for both networks separated by a DMZ for better security. Secondly, the implication of a DMZ is clear; an acceptable tradeoff involved here, is that it would be preferable to have a machine that is a more "attractive" target hacked into, for example, the Web server, that may be re-assembled in a few minutes, than it is to have the workstations or local servers that often contain a company's strategic information hacked into. There is a catch however, that with such a solution, because it presents an essential flaw, namely that of a lack of separation between servers and workstations across a private network, insider attacks are more likely to occur or, an intruder may use an internal workstation as a jumping off point for an attack, for example, by email. To avoid this, internal servers should be isolated by extra internal zones protected by a firewall (or more firewalls if so required), as illustrate in figure 3.2.

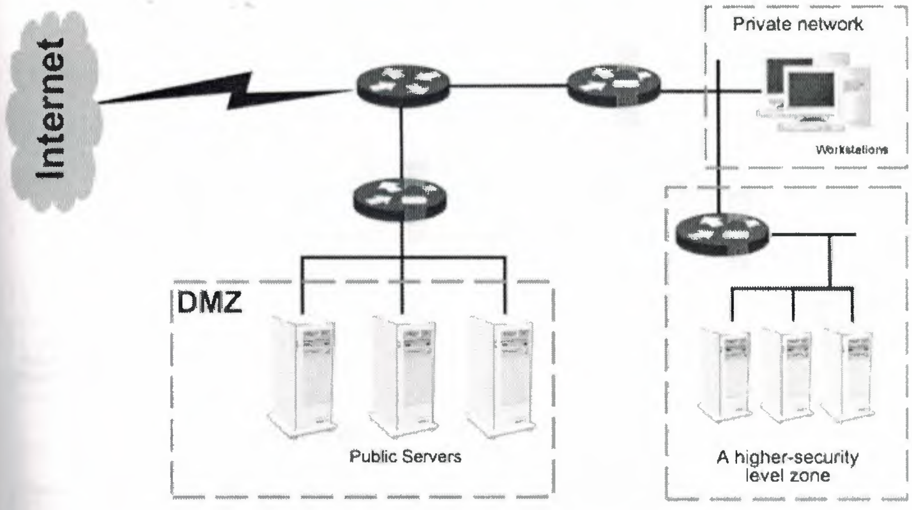


Figure3.2 Isolated Server by Extra Internal Zones

Such solutions however, are seldom used due to a poor cost-to-benefit ratio. For the servers in private networks to operate effectively, they must be appropriately protected, whilst a consistent security policy should make it impossible to get into protected areas by unauthorized users. In addition, any attempts to break into a private network could be simply detected and restrained using administrative and legal measures. The approach described above seems to be a reasonable means of providing segregation and protective isolation between various internal departments of a large organization, for example to "isolate" a research center in order to protect the research results from being captured by competitors or in large private networks such as academic and corporate networks. Here, the approach is based on physical separation of network boundaries. Figure 3.3 below illustrates an example of this type of network.

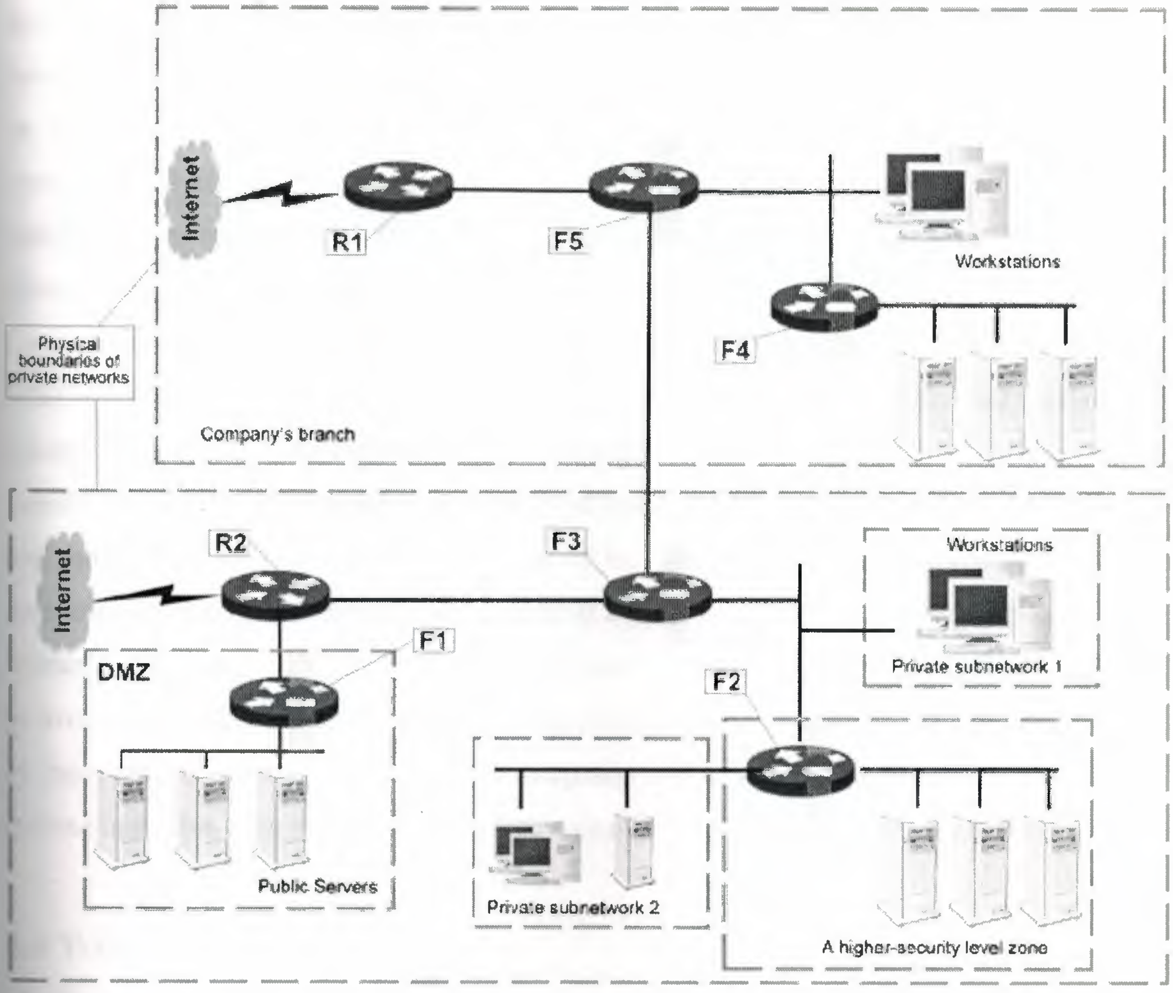


Figure3.3 Physical Separation of Network Boundaries

The R1 and R2 are perimeter routers of a private network. The objective here should be to distribute tasks between different devices (following the philosophy: “less components, less prone to damage”), let’s say, the initial packet filter can (or even should) be made only on the perimeter router, regardless of whether other protective provisions have already been implemented. Also, a dynamic NAT may be deemed necessary to sit on this device (although not always feasible). F1 – a firewall, that establishes the DMZ access rules where public servers sit. F3 and F4 are provided for dual purposes. First of all, they define a set of rules that control traffic between a private network and a public network moving in either direction. These firewalls provide VPN support for interdepartmental connections. Physically it may be a pair of copper wires, leased from an ISP, a wireless connection or any other means. Also, physical boundaries between private networks are defined by these firewalls. F2 and F5 firewalls perform similar functions within the local networks that they have been installed – they establish rules of internal server access to be followed by private subnets. Additionally, the F2 is to eliminate unuseful traffic between the subnets 1 and 2. These examples do not pretend to be models to follow in building a private network. They are merely some criteria for weighing the choice of firewall application. The reality is that this is a security policy decision first, and a firewall implementation (if at all) issue second. The above solutions still do not define what types of firewalls are to be installed across a network. Selection of firewall type and locations should also be consistent with a comprehensive security policy. Finally, the benefit of any firewall depends upon a critical issue that is common for all applications, and which may compromise the reliability of the network as a whole. Typically these solutions are enough but not always perfect: if a public network or a specially protected subnet ceases to be reachable even for a little while, the firewall application fails. In order to avoid this, redundant systems are used by configuring these systems so that, either all of them control both the incoming and outgoing traffic simultaneously or so that they resume operation after receiving a message signaling a failure of the primary system.

3.4 What is a Firewall?

My first stop is Webster's Dictionary: "A firewall is a fireproof wall used as a barrier to prevent the spread of fire. "A firewall is a system or group of systems that enforces an access control policy between two networks.". Things that well-behaved firewalls can do:

- Restrict inbound and/or outbound network traffic, based on various identifiers
- Send smoke alarms
- Log traffic (both accepted and rejected)
- Perform centralized administration for remote network access
- Provide a permeable membrane

3.5 What are the Typical Firewall Components ?

When vendors talk about firewall solutions they typically categorize the functionality into three groups:

- Packet Filter
- Circuit Level Proxy
- Application Level Proxy

Before we look into each of these areas, it's important to understand some of the basic characteristics of network packets. Believe me, there is more information stuffed into these little devils than there are college students in a phone booth, but all we really care about is:

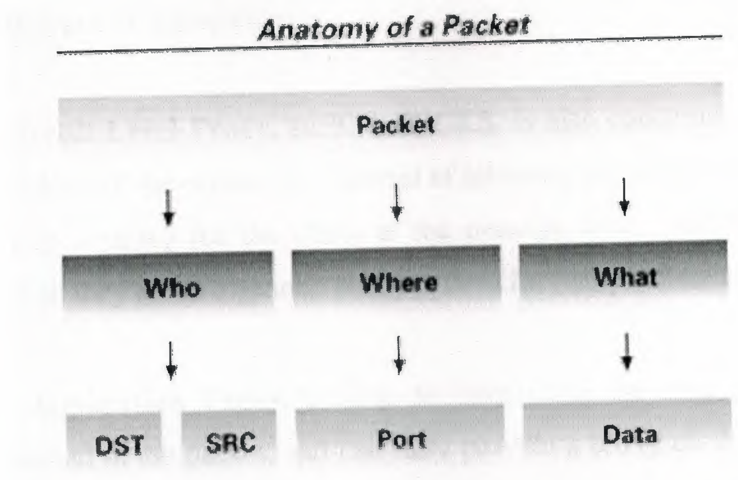


Figure3.4 Basic Characteristics of Network Packets

1. **Who** is having the conversation ?
   This is represented by a Source and Destination IP address (or ultimately a MAC address)
2. **Where**, or on what channel are they having this conversation ?

This is represented by a Port address.

3. **What** are they saying in the conversation ?
   This is represented by the Data portion of the packet.

Leave all of the other pieces in there for the eggheads who view packet analyzers like we do sitcoms. Now that we know what we are looking for in each packet, let's look at how each of these services analyzes the packet. We'll start with the packet filter.

- **A Packet Filter** only addresses who is having the conversation, and which channels they're using. It does not have the intelligence to look at the data portion. Many routers have this capability built right in and can restrict and pass traffic based on rules, addresses, and port types. The cool thing about a packet filter is that each client does not have to know where it exists in the network. It is typically placed in line of routed traffic. This saves the trouble of having to configure any information on the clients, and packet filters can be used with many types of applications.

- **A Circuit Level Proxy**, such as SOCKS, is also concerned only with the who and where of the packet. But instead of allowing the traffic to pass through, it can provide a proxy for the client at the network level. SOCKS servers are cool because they act as a generic proxy system for many different applications.

- **An Application Proxy**, is able to understand the data portion, or what is contained in the packet, and can fully provide a proxy on the client's behalf. An example of an application proxy is the HTTP proxies that many companies use to provide connectivity to the Internet. In fact, you are most likely soaking in it right now if you are reading this on the 'net! Your system is connected to an HTTP proxy, which has cached this document and you are now reading it from the cache. All of these systems used rules to make their decisions. No, not the rules that our parents used to give us like "be home at 11" and "brush your teeth before bed" these rules are usually based upon the who, where, and what of the network packet, and are programmed by the administrator. Some of the terminology used in the rule sets is strict, like Deny and Permit Access. Actually, my Dad used to talk to me like that but that's another article.

So, to summarize, application proxies proxy at the application level; a circuit level proxy proxies at the network level, and a packet filter restricts at the network level. So, we can know who is talking to whom, where they're having the conversation, and what they're saying to each other. (Where were these guys when we were passing notes in class?)

3.5.1 Typical Configurations

There are three basic configurations that are used as a base in securing a given network.

- Dual Homed
- Screened Router
- Screened Subnet

The Dual-Homed configuration is very simple, typically implementing two network cards to block or filter traffic. This machine may act as a simple packet filter or a very robust application level proxy, such as a Notes Passthru Server.

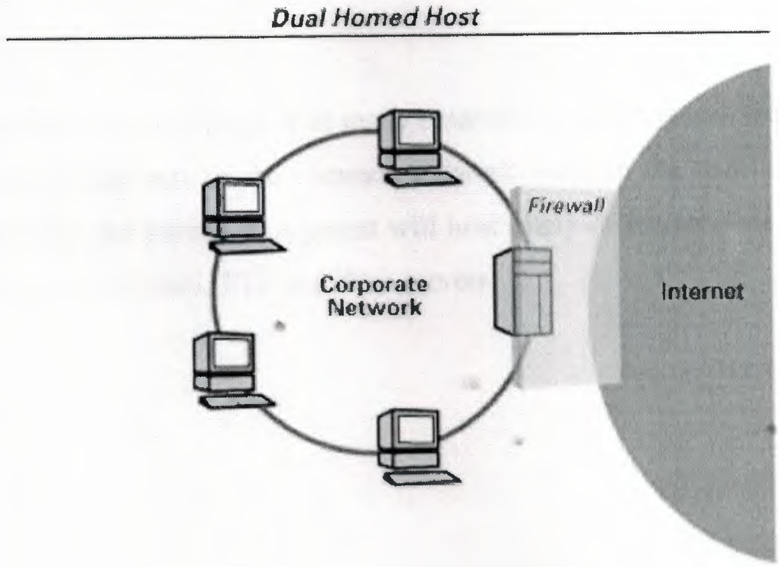


Figure3.5 The Dual-Homed Configuration

A Screened Router configuration allows only selected systems to communicate to the remote network via the router. This is typically based on a set of rules installed by the administrator.

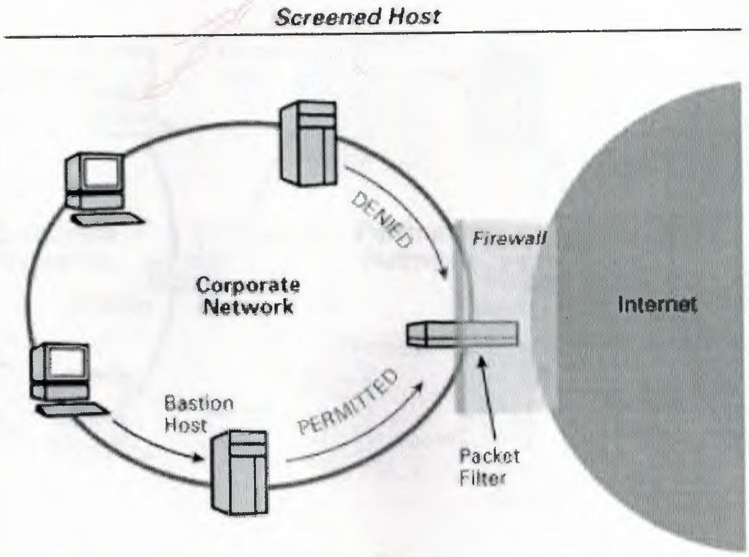


Figure3.6 A Screened Router Configuration

The Screened Subnet is more popular in many networks, and introduces the concept of a perimeter network. This acts as the common network between the two communicating networks. Typically, the perimeter segment will host many of the services that are used by both networks such as mail, FTP and Web servers.

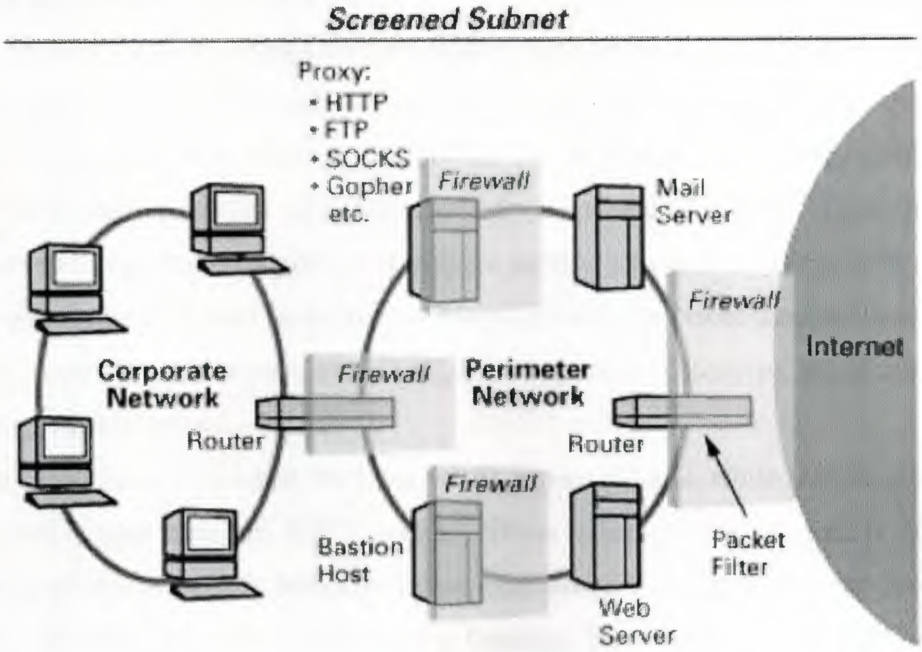


Figure3.7 The Screened Subnet

So which one do you use? Well, that's for me to know and to you to figure out! Not really. Every company is different and the security policies you have defined will dictate your eventual configuration. At the same time, lots of technologies are starting to morph into one another, so the resulting hybrid technologies can represent the best approach. There is just no turnkey information on the basic concepts, terms and designs used in firewall configurations.

3.5.2 What about Notes and Domino?

All right, keep all of the things we have talked about up to this point on a salad dish in the left portion of your brain, and let's get to the real beef here. Lotus' new Domino Web applications server uses standard HTTP, so that any browser can read published data from a Domino server. The data is dynamically converted to HTML format upon request and served to the requesting client. If you are serving up native Notes as well as Domino documents, you will need to know that Notes servers use Notes Remote Procedure Call (RPC), while Domino servers use HTTP. This important distinction needs to be factored into your firewall plans.

Native Notes has a registered Well Known TCP port of 1352, while Domino's interface is accessible via the standard HTTP port 80. These values play a key part in helping you understand how to identify Notes traffic on your network. Consider a house with different rooms; one for native Notes and one for Domino. They share the same IP address, but have different port numbers.

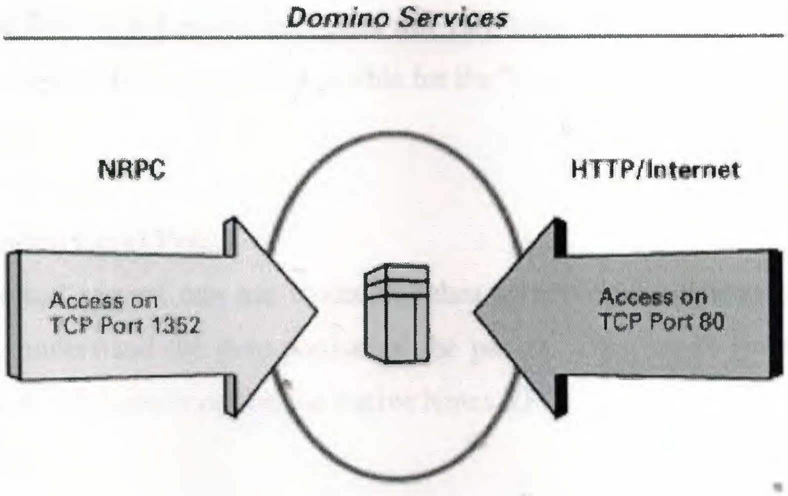


Figure3.8 Domino Servers

Now let's look at how Notes and Domino work with the different types of firewall solutions.

3.5.3 Packet Filtering

Is one of the simplest forms of firewall protection you can use. It is very common for administrators to allow only certain types of traffic through a router. For instance, you may choose to only allow TELNET (port 23) to pass through the router and restrict all other traffic. When a Notes client or server requests a connection to a destination server over IP, it will include the server's name, an IP address, and the TCP port of 1352. If you place a packet filtering device between the two Notes nodes that need to communicate, the filter will have to allow this port to be passed in the direction of the request. This support does not require any specific configuration on the client or server.

3.5.4 Circuit Level Proxy

Notes clients and servers can work with SOCKS servers.When passing through a SOCKS server was a requirement, Notes clients and servers could utilize SOCKS services by using TCP vendor stacks that support SOCKS transparently for all applications. we directly support the SOCKS 4 standard from within the application. In a sense, the application is now SOCKSified and does not rely upon specialized TCP/IP stacks to provide this support. This feature is available for the Notes client, native Notes server and Web Navigator.

3.5.5 Application Level Proxy

Notes clients and servers can use Notes Passthru servers as application proxies, since these servers understand the data portion of the packet. They speak Notes. This is the only application level proxy option for Native Notes RPCs.

3.5.6 HTTP Proxy

Notes clients and servers can also utilize HTTP Proxy servers via the HTTP Connect Method as defined here (http://home.mcom.com/newsref/std/tunneling_ssl.html).we now support the SSL Tunneling specification, which allows the native Notes RPCs to communicate through an existing HTTP Proxy. Bottom line, you can now leverage you existing HTTP Proxy infrastructure when communicating with native Notes RPCs.

3.5.7 Passthru

Consider passthru a Notes client. Since passthru is a Notes RPC application proxy, it is very robust on its own; however, support can be augmented by adding packet filtering, and other native Notes RPC proxy support mentioned earlier.

3.6 Summary

In this chapter we discussed the firewall architecture, its benefits and risks and in the last the firewall components.

4. ACTIVATE NETWORK SECURITY

4.1 Overview

Active Network Security is comprised of a number of techniques that address this shortcoming. The goal is not only to reduce the number of successful abuses of a system, but also to give early warning of abuses in progress. Finally, the objective is to ensure that misuse of the system does not go unnoticed that, should all of the security mechanisms fail, a record exists to allow corrective action.

4.2 Active Security Mechanisms

Active network security, as described in this document, encompasses networking tools and systems that allow system administrators to observe, inspect and improve the security of their networks. Many conventional security mechanisms are effective in enforcing security in a system, but lack the responsiveness necessary to maintain security on an ongoing basis. In recent years, a number of security tools have been developed that may best be classified under this heading: while these tools often have no direct effect in preventing misuse, they allow administrators to improve the overall security of their systems. Examples include:

- Intrusion Detection Systems (IDS) Intrusion Detection Systems monitor the state of a system, attempting to recognise and report improper behavior. These systems protect a network in much the same way as security cameras protect buildings: by letting security personnel keep an eye on what is going on.
- Network Security Scanners Security scanning systems inspect a network or host system, looking for known weaknesses and possible misconfigurations. The best known example is probably the Satan system it scans hosts and connected networks for a specific series of weaknesses, reporting any found, and suggesting solutions.
- System Integrity Checkers Many of the ways in which systems are attacked involve changes to the host's software and data. Integrity checkers compare the contents of a system to a known safe state allowing administrators to know exactly what has been changed.

- Honeytrap systems If an IDS is a security camera, this is a burglar alarm; systems whose sole purpose is to be attacked. By closely monitoring these systems, network administrators can observe attackers in action – allowing them to repair, learn and strengthen security against future attacks.
- Special purpose tools Specific tools have been developed to address security weaknesses present in systems. While not as generally applicable as those listed above, still deserve a place in every security administrator's toolkit. In Section 8, we will touch on two examples: password cracking systems and sniffer detector software. In a world where security mechanisms were infallible, none of these systems would be necessary. In fact, none of these systems can, in itself, prevent an attack from succeeding. The function of these tools is to minimise the effect of an attack, mitigate resulting damage, enhance the effectiveness of other mechanisms, and ensure that future similar attacks do not succeed.

4.3 The Limitations of Static Security

4.3.1 Authentication

The core of many current security mechanisms, authentication encompasses the technologies used to identify and verify the authenticity of users, network components and processes. This ranges from simple password based schemes through to biometric and cryptographic mechanisms. The ultimate goal is to uniquely associate an entity external to a system with an identity stored inside the system. In most systems, this is done by requesting some identifying information from a client, for example a password, biometric reading or response to some challenge. This information is then verified against information held inside the system. Should the identifier and stored information match, the user is authenticated; otherwise the user is denied. Extensions of this scheme include the addition of timing or locality information in the identification data, and encrypting the dialogue all aimed at making the synthesis of a counterfeit identification token more difficult.

4.3.2 Cryptography

With the recent increase in dependence on shared resources, especially public networks, the security of information in storage and transit has become a concern. Strong authentication may prevent active use of restricted resources, but passive interception of information can be as great a risk. In addition, where information is held in an untrusted system, ensuring that data remains unchanged in transit is also a concern. Cryptographic techniques are becoming increasingly prevalent in resolving these issues: ensuring that only authorised users can interpret sensitive information (encryption); and ensuring that vulnerable information is communicated intact (authentication). Encryption is the process of applying a transformation to data that can only be reversed using secret information. Depending on the application, one of two forms of encryption may be used: secret-key cryptography, where the transform and its reverse make use of the same secret, and public-key cryptography, where the encrypting transform does not require the use of secret information. Public key cryptography bears a close resemblance to the authentication problem: a user may be defined as anyone capable of reversing a given transform, thereby authenticating a communication partner. Cryptographic authentication involves the derivation of a message signature from a message, based on the use of secure hashing techniques. Should the message be modified in transit, the signature and resulting message will no longer match. In order to ensure that the message signature is not modified, encryption techniques are used (restricting the set of users capable of generating a message to those sharing a specific secret). In the case of a modified message, it is infeasible to generate a new encrypted signature that would decrypt to validate that modification. Therefore, if the signature matches the message, it is unlikely that the message was changed or counterfeited.

4.3.3 Access Control

Authentication verifies the internal identity of external parties. Access controls define which resources those parties have access to – limiting the capabilities of those users. These controls are no stronger than the authentication mechanism underlying them, and have potential weaknesses independently of authentication failure.

4.3.4 Firewalls

While firewalls could be considered a specific application of the mechanisms described above, they form one of the main pillars of current network security, and merit separate consideration. The function of a firewall is to separate networks with different security needs and policies in the most general case, to separate the internal, controlled network and any external public networks. Effectively, a firewall acts as a filter on network traffic controlling what goes into or comes out of a network.

4.4 What Do Static Methods Offer

The static methods described here, perfectly applied, are effective in ensuring the security of any network. Even in realistic environments, static security mechanisms are capable of significantly improving the security of networked resources.

- Static mechanisms can increase the security of networks in the context where they apply.
- These mechanisms can increase the technical expertise and resources required to compromise the security of a network.
- Static methods can reduce the range of attacks that Active Security mechanisms must deal with.
- Static methods can combine with Active methods to provide a synergetic improvement in security.
- Static methods can prevent attacks from succeeding.

4.5 The limitations of Static Security

In spite of the wide variety of security mechanisms available, intrusions continue to occur. Based on this fact, a number of limitations in static security mechanisms can be identified:

- The protection offered by these mechanisms is limited in scope. While these mechanisms may be effective in the context in which they are applied, they do not offer universal protection. For example, firewalls, while being effective against external attack, offer no protection against internal abuse which, as shown in a previous section, is a significant risk factor. The same type of argument applies to other mechanisms: authentication is vulnerable to trust networks, where the authentication mechanisms are bypassed. Encryption only protects information while in an encrypted form. All of the current static mechanisms can be bypassed, negating their effect.
- The security mechanisms themselves are sensitive to technical and implementation problems. Such systems can become vulnerable due to theoretical advances (such as the DES encryption standard, which can no longer be considered completely secure), or poor implementation (for example Microsoft PPTP).
- Even if theoretically sound and correctly implemented, security mechanisms must be correctly applied in order to be effective. Describes an organization that had its web server defaced while their firewall was hidden deep inside their network, acting as a log server. Many of the security mechanisms available are very complex (both in structure and in application), and a single mistake may be enough to nullify the efficacy of the system. An example of this is the use of dial in lines allowing direct access to a trusted network. No matter how good the firewall blocking official connections to that network is, it is still vulnerable.
- Static security mechanisms, by their very nature, are prone to silent failure. Often, the first sign that your security has failed comes when it is far too late (such as when an entire server is wiped clean an effective method for an intruder to erase a history of his actions). Even when a system's security has not yet been penetrated, that may lead to a mistaken sense of security. In general, these

mechanisms also cannot recognize when they are under attack – at best, an attack is logged as a series of failed transactions.

- Associated with the previous point is the issue of remedial information. Once a failure is identified, it may be difficult or impossible to trace the cause of that failure. Information on the identity and methods of an intruder may allow the effects of an intrusion to be mitigated but none of the mechanisms described offer any such capabilities inherently. The audit information collected by some tools, while being useable, does not have sufficient detail.
- Finally, the security mechanisms can themselves be subject to attack. Authentication servers can be corrupted, firewalls crashed or circumvented, and cryptographic distribution channels can be compromised. In many cases it is a simple exercise to disable system by attacking its underlying infrastructure. A good illustration of this is the number of tools that are freely available, aimed at allowing users to circumvent the restrictions applied by security mechanisms anonymous proxies, network tunneling applications and the like.

The essential problem with many of the mechanisms listed above is that they are essentially passive. While this may be sufficient for a degree of security, it does not hold up in the imperfect world of modern networks, where network administrators are often over-worked, do not have the necessary specialized skills, and where the attacks on networks are ever-escalating in complexity and intensity.

4.5.1 Sources of Attack

(a) Script Kiddies

This is the name given to the masses of relatively unskilled hackers that use the tools written by others, without necessarily having any real skill. They are typified by having endless time to spend probing networks for victims to their latest exploit tool9 – it is on these that the common perception of hackers is based. This is not to say that they do not pose a risk, however – far from it. These hackers often have an array of tools available, and keep up to date with the latest new exploit software that becomes available. In addition, since they often have no specific aims in mind (beyond the trophy of having hacked a system), they will not necessarily target the most visible or valuable machines – obscurity is no deface.

(b) Employees

Possibly the most dangerous group of potential attackers are the very people who use the networks every day the staff. They know what in a network is of value, what defences are in place, and have a ready foothold from which to escalate their control. It is a telling statistic that, in the CSI/FBI survey , 86% of respondents consider disgruntled employees as a likely source of attack (compared with 74% for independent hackers). Also, recall that 55% of respondents reported inside abuse of their networks.

(c) Mistakes

Not all anomalies in your network have hostile intent. Many "attacks" might be result from a lack of user expertise or from simple user error. This is does not imply that such errors are not dangerous: the case of the 1980 ARPAnet collapse is a clear example of how devastating a simple mistake can be.

(d) Automated Agents

This category includes such things as worms (such as the infamous 1988 Internet Worm), automated hacking tools, viruses, and Trojan software. There does not need to be a human active in order to attack systems a good example of this is the recent Melissa macro virus. With minimal modification, the Melissa virus would be capable of sending whatever document is being worked on to an email address effectively leaking information.

(e) Expert Hackers

A number of expert hacker groups have been in the media over the past few years as government witnesses, software developers, and network security experts . These groups do not merely use exploits written by others; they produce tools of their own12. They constitute the highest skill level that network security will be faced with; an administrator can expect to see completely new attacks, if any signs remain at all. The reason behind a given attack may differ wildly: recreation, industrial espionage, fraud, and attempts by foreign governments to destabilise national infrastructure have all been proposed as causes for intrusions. To place this discussion into context, consider some specific reports:

- However, the hackers of the cases on which this paper is based are known. All of them were male, and computer science students doing their master's. They all had access to the Internet, and were reasonable well acquainted with UNIX. All of the hackers, except one, had the level of an ordinary UNIX programmer with a little bit more understanding of network software.
- A sixteen year-old from the U.K. entered a plea bargain and paid a $1900 fine while another twenty-two year old pled not guilty and was acquitted on all charges in February 1998. The 16 year old was operating on a home computer in his parents' house and had a "C" grade average in his high-school computer class.
- The attackers were two teenagers from California and one teenager from Israel. Their motivations were ego, power, and the challenge of hacking into U.S. DoD computer systems.

It would appear as if the common preconception of hackers being young, male and bored holds. However, real information is scarce though a question would be whether experienced hackers get caught.

4.5.2 Outline of an Attack

The process involved in gaining control of a system generally follows a number of discrete stages, outlined below. One of the aspects that make internal abuse so dangerous is that the attacker can often bypass the early (and from an intruder's point of view, dangerous) stages, and proceed directly to escalating their control over a system13.

4.5.2.1 Exploring The Target

The first step in any intrusion is generally to build up an image of what potential targets a network contains. A number of different techniques are available to hackers, including:

- **Network Scanners**

These tools send specially constructed packets to addresses in the range being scanned. Based on the nature of the reply, it can be deduced which addresses correspond to active machines, and often even more information can be extracted: the operating system running on such systems, open ports, and the presence of intermediary network filters (such as firewalls). Detecting such sweeps has, in the past, been relatively simple: they generate a large number of similar events in system logs, within a short period of time. Increasingly, however, more complex tools are becoming effective in obscuring the details of such scans. Tools exist that allow scans to be conducted slowly, using only a few packets per hour or day or conduct a scan co-operatively from different source addresses. One common tool, allows the source of a scan to be masked by generating a number of fake scans (from spoofed addresses), and has a number of stealth scan mechanisms. One of these, a TCP ACK scan, has been found to be effective in penetrating our testbed firewall.

- **DNS Zone Transfer**

By retrieving all information available for a network from the DNS hierarchy, an attacker can retrieve a list of all externally accessible points for that network. In addition, if the internal DNS servers are accessible externally, an attacker has access to a wealth of information: a map of the host names and addresses of all machines on the network, and possibly even account details for the system maintainer.

- **Tracing the system neighborhood**

Using the DNS and addressing information and tools such as trace route, an attacker can determine what machines are in a network neighborhood. Compromising a machine on the external path of a target network, a number of attack forms become available – ranging from simple traffic snooping to TCP session hijacking . Compromising a machine that the target network depends on, such as a DNS cache server, similarly opens the door for attacks on the target network – and that machine may be significantly less secure than the protected network .

4.5.2.3 Penetration

The goal of this step is to gain an executing process on the target system. A vast number of exploits are known (with more being discovered every month) allowing an unauthorized user to gain a foothold on the victim host. Examples include server buffer overflows, system backdoors and weak authentication or access control mechanisms discusses some specific examples of well known attack techniques. It is this phase that IDS attempts to recognise – therefore it is also at this point that monitoring systems are likely to be attacked. Using denial of service (DoS) attack, or customised exploits, an attacker may attempt to disable the security mechanisms in a network. Alternatively, an attacker would use his knowledge of the organisation's traffic patterns to hide the attacking traffic in normal traffic streams – making filtering and detection more difficult. For example, a CGI exploit disguised as a normal HTTP request is likely to bypass any filtering mechanisms in place (as demonstrated in the firewall experiments).

4.5.2.4 Escalation

Once an attacker has a foothold on a system, the next step is to escalate to control over the system. In this step, the goal is to gain sufficient administrative privileges to allow the next step, Embedding, to proceed – or to do damage. This often takes the form of a bootstrapping process: initially, the attacker starts with minimal privileges. Then, using a succession of exploits and attacks, an attacker gains successively greater privileges until he has complete control over the system. Alternatively, this could be bound to the Penetration step: many services run with extensive privileges, and grant an attacker those privileges when compromised (effectively allowing an attacker to bypass this step which is why most services run with as few privileges as possible).

4.5.2.5 Embedding

Having gained control of a system, an attacker will cement his control over a system, so that later intrusions do not require the dangerous Penetration and Escalation steps to be repeated. This step involves removing all records of the initial intrusion, bypassing or disabling the reporting mechanisms, and building access routes that will allow the attacker to resume control of the compromised system at a later time. This ensures that the attack and access routes are not detected ensuring that backdoors remain accessible. Examples of embedding techniques include: modifying access control files to allow the attacker access (e.g. adding accounts to a system); modifying access control mechanisms so that they do not apply to the attacker (e.g. adding a master password to the login program). Another mechanisms is to place tools that allow rapid escalation into low-privilege accounts (and ensuring that those remain accessible) these may be harder to detect. An example of this method is the placement of SUID-root command shells (under Unix) – allowing the user to instantly gain complete control over a system. A final mechanism is placing a server process on the machine that will accept commands from the attacker.

4.5.2.6 Extraction

At this point, the attacker has effectively gained complete control over the system. In many cases it is at this point that an attacker would extract information from the system, or attack the information held on the system (such as vandalising a web site hosted from a compromised server). Security systems such as firewalls may no longer hinder an attacker many techniques exist for communicating invisibly through filtering systems.

4.5.2.7 Relay

Once an attacker has completed modifying or extracting information from a system, he will often retain that system for use as a springboard for further attacks. Tracing an attacker backward through the complex interconnected networks available is very difficult attackers make use of multiple systems to obscure the true source of attack. In addition, tools are emerging that allow distributed attack and scanning of systems – not only obscuring the attacker, but making the attacks harder to detect and counter. An emerging trend is for attackers to target home machines permanently connected to the Internet. Such machines often have very low security, and are ideal as staging areas for further attacks. Who would be liable for damage done from such a compromised machine is unclear what is clear is that systems need protection, whether or not they contain critical resources.

4.6 Typical Attack Techniques

- **Scanning a network**. The first step in an attack is reconnaissance – finding out as much as possible about the target. Many tools are available for investigating a network – ranging from simple scripts to commercial network mapping tools, to dedicated scanning applications19. In essence, these tools send a packets to a potential host, and deduce information about that host from any reply. Mapping a network consists of checking every possible address for that host. In particular, a number of scan types can be distinguished.
- **Ping scan**: The simplest form of scan, an attacker sends an ICMP echo request packet to every candidate machine (which is the same way the ping tool works). Any addresses that respond are noted as active.
- **TCP Connect() scan**: Another simple scan, an attacker attempts to open a standard TCP connection to a typical port on the candidate machine (such as the HTTP port 80). Any machine where such a connection succeeds is noted as active. Since many systems log any connection attempts, this type of scan is relatively easy to recognise from standard audit data.
- **TCP SYN (Stealth) scan**: This scan sends a connect request to every candidate machine (similar to the Connect() scan), but does not complete the connection by sending a final SYN/ACK packet. In this way, the connection fails and does not generally show up in the system logs – hence a “stealth” scan. Since this scan has a similar signature to a SYN flood attack, many security systems now log such occurrences.
- **Stealth FIN, Xmas, ACK and NULL scans**: These scans all form part of the same family of variations on the SYN scan techniques. Each sends a special packet to a candidate address, deducing whether a port is open or not from RST reply packets (which indicate a closed port). If not reply is received the port is open – or the request lost in transit, such as being discarded by a firewall. FIN scans consists of packets with the FIN flag set, Xmas scans of packets with the FIN, URG and PUSH flags set, and NULL scans of packets with no set flags.

The ACK scan consists of packets with the ACK flag set (generally denoting replies), and so are often capable of penetrating firewalls .

- **UDP scans**: This scan consists of sending UDP packets to likely ports on candidate machines at worst, scanning for any open UDP ports. Since UDP is connectionless, such attempts are harder to control using filtering firewalls, and may be capable of finding unprotected services and hosts. Many variations on these scanning techniques exists – including scans using fragmented packets, and scans spread across a long period or a number of source machines. In practice, completely blocking scans is probably infeasible – but may give an administrator early warning of an impending attack.

- **Buffer Overflows**. This is actually rich category of specific attacks, all using similar weaknesses in software. The core of the attack is to pass an unusually structured (often very long) value as a parameter to a system, when it is expecting something else – for example, requesting an FTP server to change the working directory to an extremely long filename. What happens, in general, is that the parameter overflows its storage buffer, overwriting commands that would later be executed – allowing an attacker to have arbitrary commands executed by the remote server. These commands can then be used to do any number of things – typically, creating an interactive shell, modifying access restrictions, or retrieving sensitive information, such as a password list. for details on this technique.
- **Open doors and abused trust**. In order to simplify authentication and access control, many systems accept assertions made by trusted systems. For example, the rsh series of commands accepts the remote machine's claims to user identity, if the remote machine is authorised to make such claims. This allows a number of attack techniques, based around abusing the assumptions made in such systems. One technique involves an attacker assuming the identity of a trusted machine, allowing it access to the trusting system. Another is based on the fact that under some systems (such as some Unix variants), users can control which other machines are trusted (using the .rhosts file). A common escalation step in

attacking such a host is to modify this file, to allow the attacker free access. For an example of the process involved.

- **Social Engineering**. This type of attack is one of the oldest, and most effective way of bypassing security mechanisms: fool somebody with the ability to do it for you. Variations range from guessing information based on the attacker's knowledge of the target involved, to impersonating personnel, and more. The only way to protect an organisation is to ensure that it has a sufficiently clear security policy, and that its users are educated – no technical measures can prevent this type of attack. For a good example of how effective this can be.
- **Application Attacks**. These attacks depend on convincing an application to do something it was not expected to – overwrite files, execute commands it should not, or give away information that should be hidden. In addition, these attacks are notable since they can often penetrate even the best developed security mechanisms – the only defence is to keep the applications themselves secure. Examples include requesting password files via FTP or HTTP, attempting to overwrite sensitive files via the same, or passing unexpected information to server applications – such as any of the range of CGI exploits available. For a good example of how this type of attack proceeds.

- **Trojan software**. The problem of computer viruses is well-known; but the techniques used for propagating these programs can also be used to compromise security. A good example is the Back Orifice system – once an infected application is run on a system, it installs a backdoor on the system, allowing the attacker free access. Preventing this type of attack is difficult – it requires user education, and security to be deeply embedded into systems.

4.7 Policy Issues for Active Security

4.7.1 What is Security Policy?

An organisation's Security Policy defines and outlines the measures present to ensure that the confidentiality, integrity and availability of systems remain intact20. This includes such items as:

- **System review**: What systems are in place and in need of protection.
- **Risk assessment**: What the risk factors affecting such systems are, and how vulnerable the organisation is to harm should one of these risks be realised.
- **General intent**: How the policy is to be interpreted, and how to resolve issues not directly covered in the policy.
- **Measure selection**: A listing of what measures are in place, describing their placement, configuration, and operational parameters.
- **Operational protocols**: What steps are to be taken under specific circumstances, such as system update protocols and change management, intrusion response and general operations.
- **Responsibility allocation and authority**: Who is responsible for specific actions or parts of the systems, and what authority they bear.
- **Security policy information**: When and how the policy is reviewed, where it is kept, and what authority underwrites it.

In effect, the security policy of an organisation circumscribes the measures taken by an organisation to ensure that computing systems are protected under operational and adverse circumstances. Two main techniques are generally used to ensure that resources are adequately protected: baseline protection and customised protection.

Baseline protection implies the application of security mechanisms across the entirety of a system or subsystem, without regard for the specific needs of components. This requires minimal risk assessment, and may offer acceptable security in low-risk environments, but generally will not offer the most cost-effective protection or adequately protect sensitive systems. In addition, certain safeguards may actually reduce the security of a system (in terms of the critical factors mentioned above). For example, encryption improves the confidentiality of systems, but decreases availability. Therefore, for systems where high availability supersedes confidentiality (e.g. internal email systems), the use of this mechanism reduces overall security. Customised protection is the application of security mechanisms based on a detailed risk assessment, in order to address the particular needs of a system. This ensures the most efficient allocation of resources, and avoids the problem of inappropriate security measures, but requires a more complex assessment of the needs of an organisation. In addition, an incomplete assessment would result in a mismatch between the actual and estimated needs of a system, creating gaps in the security present. A method that is often used is to combine the techniques described above: using baseline security to increase overall protection, and protecting critical or sensitive systems with custom measures. This offers many of the advantages of both worlds: a common base of protection system-wide, sufficient protection for vulnerable systems, protection against changes in risk patterns, and simplified administration. Intrusion Detection and Active Security mechanisms lend themselves to both baseline and customized security. Applying these measures system-wide allows the system to be protected against general misuse, but may require significant resources. By optimising the placement and configuration of these tools, it is possible to offer both increased protection for sensitive systems, and more context-sensitive detection, at the cost of general protection. For example, IDS deployment often concentrates monitors in high-risk areas, such as network ingress points (e.g. adjacent to firewalls), or in the presence of valuable resources (such as network server farms).

4.7.2 The Relationship between Active Security and Security Policy

The Active Security tools discussed in this document are capable of being used as part of a baseline security strategy. This is also effectively what an organisation defaults to, when no formal Security Policy is set out. In order to be used to greatest effect, however, these tools need to be deployed and configured with knowledge of the needs and behaviour of the specific systems involved. As an illustration, IDS can function on any network or host system, attempting to recognise generally known abusive behaviour (such as invalid network traffic). Such a system will not be capable of recognising misuse, where such misuse does not correspond to anomalous or illegal activity. For example, such an IDS would offer no protection against users attempting to access resources in an inappropriate manner: for example, Joe from Sales attempting to read the personnel database (using a syntactically legal query). Embedding information from the security policy into such tools can greatly improve their efficacy. To extend the example, if it is known that certain actions are precluded by the security policy, the IDS and other tools could be configured to include this information. Knowing that nobody outside the personnel department can access that database, an IDS could easily detect Joe's attempt. The IDS can report the problem to security personnel – whether this is a case of internal abuse, or Joe's identity has been compromised and abused. In addition, Active Security tools can only function correctly if they are constantly maintained and monitored. As such, they depend on a security policy that defines how, and by whom, they are to be cared for these tools rapidly lose their function if they are ignored. As described more fully in the next section, the reporting capabilities of these tools also imply the need for policies to be set out, in order to handle the changing system. The security policy may also develop from the results gained from Active Security measures. These tools offer rich detail on the security state of a system: which areas are weak, which areas are being attacked, and the general behaviour of a system. This allows the system administration to extract system-specific information on the real security needs of the system, and modify the security policy accordingly. The information gained from these tools can show not only security problems – but also performance, management and configuration problems, and may give early warning of system failures.

4.8 Tools Supporting Active Security

4.8.1 Network Mappers

A variety of commercial and free network discovery tools are currently available examples, These tools use many of the same techniques described in section 3 to explore the content of networks: DNS zone transfers, scanning the address and port space, requesting information from hosts found, and promiscuous monitoring of a network. In fact, many of these tools are now used by attackers nmap, for example, was an invaluable aid in inspecting the exact coverage of the firewall policy during our experiments.

As an example of how a typical network mapper works, consider the nmap tool. It is a powerful aid in exploring networks not only because it offers a wide variety of scanning options but also due to its unique ability to identify a wide variety of hosts systems, down to the operating system, and sometimes version. Nmap works by sending packets with a wide variety of special characteristics to hosts being investigated: packets with specific (often illegal) flags set, ICMP echo packets, fragmented packets (again, sometimes with illegal fragmentation), etc. Every host has a particular style of responding to such packets – by combining these response characteristics, it is possible to narrow down exactly what system is present on the interrogated host. In fact, nmap uses a signature analysis system which bears some similarity to that used by IDS systems to recognise specific attacks – allowing the tools to easily extend its library of recognized systems. For example, it is possible to recognize Linux systems with older kernels than version 2.0.35 by the fact that, presented with a packet with the SYN flag and an illegal flag set, these systems retain the illegal flag in their response. Scanning a network generates a mass of highly anomalous packets alerting any good IDS tools present – and may have unwanted side effects. Because of the use of unusual traffic patterns, these tools are capable of damaging a network system certain types of fragmentation patterns.

4.8.2 Network Security Scanners

Configuring networks and network hosts to be secure is a difficult task: validating that such a system is secure may be even more difficult. A single security weakness in a configuration is all an attacker needs: a single weak password, a single outdated server, or a single vulnerable port. Network mapping tools go some way towards allowing an administrator to verify systems. Network security scanners (also known as vulnerability assessment tools) take this a step further – they actively test the security of a system against a number of attack scenarios, reporting on the location, severity, and solution to weaknesses found. These tools have had a contentious history – from the early COPS system, to the controversial Satan tool, to the current range of freely available toolkits, such as Nessus, Internet Security Scanner and Cybercop Scanner. Because these tools are capable of automating the vulnerability identification phase of an attack, it was felt by some that releasing such tools encourage script kiddies to attack systems. In practice, similar tools are available in the hacker community scan being a good example. Like IDS systems, these tools come in two varieties: host-based and network-based systems. Hostbased systems (such as COPS) analyse the security mechanisms in place on a system, looking for possible misconfigurations or dangerous settings. Examples include accounts with weak passwords, excessively trusting systems, and applications with unusual privileges (which may simply be a misconfiguration, or may be indicative of a past intrusion). This review is generally extremely system specific, but allows a wide range of issues to be checked across many user accounts a potentially significant saving for overworked administrators.

The second class, that of network-based systems, check hosts for secure networking policies. Tests include weak passwords for well-known accounts, the presence of services known to be dangerous (e.g. NFS available from outside a firewall), and unnecessary services (e.g. NFS without shared file systems). In addition, these tools include libraries of exploits, which are tested against subject systems checking whether such systems are susceptible to the specific weaknesses. In effect, the tool attempts to break into the subject system – if it succeeds, there is clearly a security flaw.

Finally, network-based systems are presently developing mechanisms for reviewing other security systems, such as IDS and firewalls. In particular, these systems can simulate the techniques used by attackers, allowing an administrator to verify that these are blocked or detected by the firewall or IDS, as appropriate. One issue with such systems that is sometimes overlooked is that these systems must be kept up to date constantly ensuring that a network is secure against last year's attacks does not offer any benefit against current risks. As the attack techniques used against systems evolve, these system should be updated, and the systems re-inspected.

4.8.3 System Integrity Checkers

Once a system is compromised, one of the first actions taken by an intruder involves changing system files: to disguise the intrusion, facilitate future penetrations, or support escalation in control over the system. In addition, there is a variety of events that will result in unauthorised changes to system files ranging from viral infection, unauthorised changes by administrative personnel, or failing hardware. A tool developed to address this problem is the well-known Tripwire package. It has since become a standard component in many system administrator's toolkits. In essence, the Tripwire system stores a hashed snapshot of file system features and content, compares this to the current system state, and reports any discrepancies.

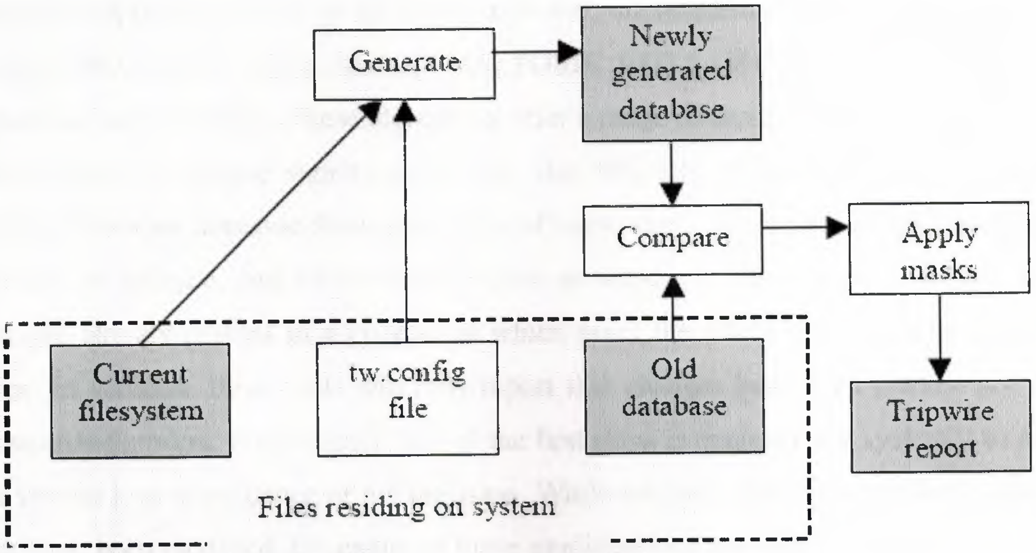


Figure4.1 Structure of the Tripwire System

As can be seen from the above diagram, a Tripwire configuration consists of two main components: the Tripwire configuration files, and a previously generated reference database for that system. The configuration files consist of a series of file or directory paths and attribute masks (defining which attributes of a file may safely be ignored), or of M4-style preprocessing commands (similar to those used by the cpp C-language preprocessor). Using these features, it is possible to create fine grained configurations with support for host-specific variations.

The reference database is generated by Tripwire, based on some initial trusted file system. It is important to ensure that this initial generation is done on an uncompromised system ideally, this should be created for a system after the initial configuration, but before that system is taken into use. Tripwire cannot detect preexisting problems – only changes that occur after its installation. The security of the Tripwire system is based on a number of factors: the integrity of the Tripwire software itself, the integrity of the reference database, and the strength of the hashing algorithms usedto identify files. Therefore, it is suggested that the reference database be stored in a secure location: on a different, secure system, or on read-only media.

To minimize the chance of an attacker making undetectable modifications to files, Tripwire supports the use of up to 10 different, simultaneous hashing algorithms (by default: MD5, MD4, MD2, Snefru, SHA, POSIX 1003.2 CRC-32 and CCITT CRC-16 signatures are available). These algorithms offer a range of security/performance features, and the use of multiple signatures increase the difficulty of generating hash collisions greatly. From an Intrusion Detection point of view, this type of tool is most useful as a last line of defence, and for recovering from an intrusion. These tools will only report changes already present in a system at which point the attack may be in an advanced stage. In addition, these tools will only report that changes have been made – not what those changes were. For example, one of the first steps in controlling a system is to purge the system logs of evidence of the intrusion. While integrity checkers may detect that the logs have been modified, the nature of those modifications may not be evident.

System integrity checkers offer a strong deterrent, and can be of inestimable value in mitigating the effects of an intrusion, but they are best suited as a last line of defence. Once an intrusion has progressed to the point where system files are compromised, much of the potential damage could already have occurred – particularly where a loss of confidentiality is concerned.

4.8.4 Password Crackers

Many modern security systems have moved away from the user password authentication scheme, using biometric identities, cryptographic schemes, one-time passwords, and the like. For the large group remaining, however, weak passwords remains a significant problem. Password crackers are tools that attempt, through a combination of social engineering and brute force, to guess the password associated with a resource [Muffett92]. These tools are well-known as a major risk in the Unix world [Farmer93], but have recently found their way into many other systems in fact, a password cracker is now available for virtually every system using key-phrase based protection, such system authentication and file encryption.

In addition, the computing power available to crackers is increasing the level of complexity needed for secure passwords current recommendations in length, and changed every 3-6 months. However, even with these recommendations in place, human nature tends to use the simplest solutions – hence the problem of weak passwords. From a security point of view, password crackers allow an administrator to identify and address weak passwords before they become a problem. On the counter side, attackers also find such tools invaluable in gaining access to systems. Running periodic checks on passwords used, especially for sensitive accounts, can make a system more secure – but is not a replacement for user education, and stronger authentication mechanisms.

4.8.5 Sniffer Detection

Many of the current network protocols were designed to function in a trusted environment. Protocols such as Telnet, HTTP, FTP, and many others carry sensitive information in clear format any person observing the network traffic can extract such information, a typical example being username password pairs on a network login. Attackers are well aware of this fact, and often place network monitoring tools, or sniffers, on compromised hosts. The traffic captured on such hosts can then be used to compromise better protected hosts, or gather sensitive information. Since there is often no need for special equipment for this monitoring, it can be very difficult to identify which hosts may be observing confidential exchanges. In response to this problem, a series of tools have been developed so called sniffer. These tools use a number of techniques to attempt to isolate eavesdroppers:

- **MAC / Protocol addressing mismatches**: Many network protocol stacks do not verify that messages received were actually sent to their addresses – they rely on lower levels in the stack for that. A machine in promiscuous mode may therefore respond to requests sent out to its correct protocol address – even if the lower level MAC address was incorrect. A machine will then only respond to such requests if the MAC address filtering is not active – or it is in promiscuous mode.

Examples of such requests include ICMP Ping, UDP or TCP echo (or other ports that always respond), or requests that generate error replies. The core of the method is attempting to fool a host into replying to a request it should not have been capable of seeing.

- **DNS Test**: Many machines automatically do reverse DNS lookups on IP addresses not yet mapped. Therefore, by sending messages to fictitious hosts, and monitoring reverse lookups, a sniffer detector can recognise machines monitoring traffic.
- **Decoy method**: Attackers will often sniff networks looking for such items as remote login sessions in the setup phase of protocols such as FTP, Telnet, or POP. By generating false login transactions, and monitoring for attempts to make use of that information, it is possible not only to identify the presence of monitors, but also to verify that these are being used to attack a network.
- **Latency tests**: In most modern networks, the resolution of MAC addresses is handled by hardware on the network interface. Therefore, the workload of a machine on a heavily loaded network segment will depend only on the traffic destined for that machine – unless it is observing all traffic. By comparing the response patterns of machines on lightly and heavily loaded links, sniffer detector tools can determine whether a machine appears to be in promiscuous mode. A machine monitoring the segment will have to interpret every message on a heavily loaded link, placing a high processing overhead on that machine. Therefore, the response pattern for a monitor will differ greatly between light and heavy loads, while for normal configurations the patterns should be near identical.
- **Direct inspection**: Directly checking the state of a network adapter on host machines is possible and may be the only way to detect which machines are in promiscuous mode under certain circumstances. This method may not be feasible on large networks, and on an ongoing basis, however. Of course, tools have been developed to attempt to avoid detection – but the presence of an unauthorized monitor on a network is a strong indication that there is a security problem.

4.8.6 Honeytrap Systems

As pointed, current IDS methodologies have a number of shortcomings, including problems recognizing novel attacks, the occurrence of false positives, and reporting of attacks that are of no interest (because the system is known to be invulnerable to these attacks). A tool which attempts to bridge these gaps is that of honeytrap systems simulated or real systems that exist for the sole purpose of being attacked. In essence, the goal of these systems is to act as bait encouraging attackers to attack these in preference to more valuable parts of a network. Once such a system is attacked, an administrator knows that the network is under attack, and can closely monitor the attacker. Since the system is not generally used, the problem of false positives does not occur any activity on that system is hostile. Since the system does not depend on recognizing specific attacks, and the limited activity levels allow thorough

review of all activity on that system, novel attacks can be observed and studied. Finally, the fact that an attacker has penetrated the honeytrap system implies that other systems on the network are vulnerable. One of the most important aspects of a honeytrap system is that it should not be recognizable as such to an attacker, it should look and behave as a real system would. In addition, in order to learn from such a system, it should be configured similarly to the real systems on a network, allowing lessons learned there to be applied directly in improving the security of more valuable machines. While software tools are available that simulate networks and hosts, in this section we shall focus on honeytraps built out of dedicated systems.

The first step issue in setting up a honeytrap is to ensure that it does not reduce the security of other systems on the network. Should a honeytrap system be compromised, it must be ensured that this system cannot be used to attack other systems (on the same network, or any other). Many of the mechanisms used in firewalls apply here aimed at keeping the intruder in, rather than out. Secondly, an administrator should ensure that the honeytrap system gathers as much information as possible. In addition, this information should be kept in a safe area since it is assumed that the honeypot will become compromised. In addition, this increased logging should be hidden from an attacker, to avoid them focusing on "less protected" and more valuable systems. For an eminently readable description of how a honeytrap works, and what its value in a system under attack. The legality of using such systems has been a subject of some discussion the conclusion of which was that a honeytrap is no more illegal than a burglar alarm.

4.9 Summary

In this chapter we discussed activate security mechanisms, limitations of static security and we identified the limitation, what static methods offer, sources of attack, typical attack techniques, policy issues for active security, tools supporting active security and in the last we discussed honeytrap systems.

5. INTERNET SECURITY

5.1 overview

As of 1996, the Internet connected an estimated 13 million computers in 195 countries on every continent, even Antarctica (1). The Internet is not a single network, but a worldwide collection of loosely connected networks that are accessible by individual computer hosts in a variety of ways, including gateways, routers, dial-up connections, and Internet service providers. The Internet is easily accessible to anyone with a computer and a network connection. Individuals and organizations worldwide can reach any point on the network without regard to national or geographic boundaries or time of day.

However, along with the convenience and easy access to information come new risks. Among them are the risks that valuable information will be lost, stolen, corrupted, or misused and that the computer systems will be corrupted. If information is recorded electronically and is available on networked computers, it is more vulnerable than if the same information is printed on paper and locked in a file cabinet. Intruders do not need to enter an office or home, and may not even be in the same country. They can steal or tamper with information without touching a piece of paper or a photocopier. They can create new electronic files, run their own programs, and hide evidence of their unauthorized activity.

5.2 Basic Security Concepts

Three basic security concepts important to information on the Internet are confidentiality, integrity, and availability. Concepts relating to the people who use that information are authentication, authorization, and non repudiation.

When information is read or copied by someone not authorized to do so, the result is known as loss of confidentiality. For some types of information, confidentiality is a very important attribute. Examples include research data, medical and insurance records, new product specifications, and corporate investment strategies. In some locations, there may be a legal obligation to protect the privacy of individuals. This is particularly true for banks and loan companies; debt collectors; businesses that extend credit to their customers or issue credit cards; hospitals, doctors' offices, and medical testing laboratories; individuals or agencies that offer services such as psychological counseling or drug treatment; and agencies that collect taxes.

Information can be corrupted when it is available on an insecure network. When information is modified in unexpected ways, the result is known as loss of integrity. This means that unauthorized changes are made to information, whether by human error or intentional tampering. Integrity is particularly important for critical safety and financial data used for activities such as electronic funds transfers, air traffic control, and financial accounting. Information can be erased or become inaccessible, resulting in loss of availability. This means that people who are authorized to get information cannot get what they need. Availability is often the most important attribute in service-oriented businesses that depend on information (e.g., airline schedules and online inventory systems). Availability of the network itself is important to anyone whose business or education relies on a network connection. When a user cannot get access to the network or specific services provided on the network, they experience a denial of service.

To make information available to those who need it and who can be trusted with it, organizations use authentication and authorization. Authentication is proving that a user is whom he or she claims to be. That proof may involve something the user knows (such as a password), something the user has (such as a "smartcard"), or something about the user that proves the person's identity (such as a fingerprint). Authorization is the act of determining whether a particular user (or computer system) has the right to carry out a certain activity, such as reading a file or running a program. Authentication and authorization go hand in hand. Users must be authenticated before carrying out the activity they are authorized to perform. Security is strong when the means of authentication cannot later be refuted - the user cannot later deny that he or she performed the activity. This is known as non repudiation.

5.3 Why Care About Security?

It is remarkably easy to gain unauthorized access to information in an insecure networked environment, and it is hard to catch the intruders. Even if users have nothing stored on their computer that they consider important, that computer can be a "weak link", allowing unauthorized access to the organization's systems and information. Seemingly innocuous information can expose a computer system to compromise. Information that intruders find useful includes which hardware and software are being used, system configuration, type of network connections, phone numbers, and access and authentication procedures. Security-related information can enable unauthorized individuals to get access to important files and programs, thus compromising the security of the system. Examples of important information are passwords, access control files and keys, personnel information, and encryption algorithms. The consequences of a break-in cover a broad range of possibilities: a minor loss of time in recovering from the problem, a decrease in productivity, a significant loss of money or staff-hours, a devastating loss of credibility or market opportunity, a business no longer able to compete, legal liability, and the loss of life.

5.4 Network Security Incidents

A network security incident is any network-related activity with negative security implications. This usually means that the activity violates an explicit or implicit security policy (see the section on security policy). Incidents come in all shapes and sizes. They can come from anywhere on the Internet, although some attacks must be launched from specific systems or networks and some require access to special accounts. An intrusion may be a comparatively minor event involving a single site or a major event in which tens of thousands of sites are compromised. (When reading accounts of incidents, note that different groups may use different criteria for determining the bounds of an incident.) A typical attack pattern consists of gaining access to a user's account, gaining privileged access, and using the victim's system as a launch platform for attacks on other sites. It is possible to accomplish all these steps manually in as little as 45 seconds; with automation, the time decreases further.

5.5 Sources of Incidents

It is difficult to characterize the people who cause incidents. An intruder may be an adolescent who is curious about what he or she can do on the Internet, a college student who has created a new software tool, an individual seeking personal gain, or a paid "spy" seeking information for the economic advantage of a corporation or foreign country. An incident may also be caused by a disgruntled former employee or a consultant who gained network information while working with a company. An intruder may seek entertainment, intellectual challenge, a sense of power, political attention, or financial gain. One characteristic of the intruder community as a whole is its communication. There are electronic newsgroups and print publications on the latest intrusion techniques, as well as conferences on the topic. Intruders identify and publicize misconfigured systems; they use those systems to exchange pirated software, credit card numbers, exploitation programs, and the identity of sites that have been compromised, including account names and passwords. By sharing knowledge and easy-to-use software tools, successful intruders increase their number and their impact.

5.6 Types of Incidents

Incidents can be broadly classified into several kinds: the probe, scan, account compromise, root compromise, packet sniffer, denial of service, exploitation of trust, malicious code, and Internet infrastructure attacks.

- **probe**

A probe is characterized by unusual attempts to gain access to a system or to discover information about the system. One example is an attempt to log in to an unused account. Probing is the electronic equivalent of testing doorknobs to find an unlocked door for easy entry. Probes are sometimes followed by a more serious security event, but they are often the result of curiosity or confusion.

- **Scan**

A scan is simply a large number of probes done using an automated tool. Scans can sometimes be the result of a misconfiguration or other error, but they are often a prelude to a more directed attack on systems that the intruder has found to be vulnerable.

- **Account Compromise**

An account compromise is the unauthorized use of a computer account by someone other than the account owner, without involving system-level or root-level privileges (privileges a system administrator or network manager has). An account compromise might expose the victim to serious data loss, data theft, or theft of services. The lack of root-level access means that the damage can usually be contained, but a user-level account is often an entry point for greater access to the system.

- **Root Compromise**

A root compromise is similar to an account compromise, except that the account that has been compromised has special privileges on the system. The term *root* is derived from an account on UNIX systems that typically has unlimited, or "superuser", privileges. Intruders who succeed in a root compromise can do just about anything on the victim's system, including run their own programs, change how the system works, and hide traces of their intrusion.

- **Packet Sniffer**

A packet sniffer is a program that captures data from information packets as they travel over the network. That data may include user names, passwords, and proprietary information that travels over the network in clear text. With perhaps hundreds or thousands of passwords captured by the sniffer, intruders can launch widespread attacks on systems. Installing a packet sniffer does not necessarily require privileged access. For most multi-user systems, however, the presence of a packet sniffer implies there has been a root compromise.

- **Denial of Service**

The goal of denial-of-service attacks is not to gain unauthorized access to machines or data, but to prevent legitimate users of a service from using it. A denial-of-service attack can come in many forms. Attackers may "flood" a network with large volumes of data or deliberately consume a scarce or limited resource, such as process control blocks or pending network connections. They may also disrupt physical components of the network or manipulate data in transit, including encrypted data.

- **Exploitation of Trust**

Computers on networks often have trust relationships with one another. For example, before executing some commands, the computer checks a set of files that specify which other computers on the network are permitted to use those commands. If attackers can forge their identity, appearing to be using the trusted computer, they may be able to gain unauthorized access to other computers.

- **Malicious Code**

Malicious code is a general term for programs that, when executed, would cause undesired results on a system. Users of the system usually are not aware of the program until they discover the damage. Malicious code includes Trojan horses, viruses, and worms. Trojan horses and viruses are usually hidden in legitimate programs or files that attackers have altered to do more than what is expected. Worms are self-replicating programs that spread with no human intervention after they are started. Viruses are also self-replicating programs, but usually require some action on the part of the user to spread inadvertently to other programs or systems. These sorts of programs can lead to serious data loss, downtime, denial of service, and other types of security incidents.

- **Internet Infrastructure Attacks**

These rare but serious attacks involve key components of the Internet infrastructure rather than specific systems on the Internet. Examples are network name servers, network access providers, and large archive sites on which many users depend. Widespread automated attacks can also threaten the infrastructure. Infrastructure attacks affect a large portion of the Internet and can seriously hinder the day-to-day operation of many sites.

- **Incidents and Internet Growth**

Since the CERT® Coordination Center began operating in 1988, the number of security incidents reported to the center has grown dramatically, from less than 100 in 1988 to almost 2,500 in 1995, the last year for which complete statistics are available as of this writing. Through 1994, the increase in incident reports roughly parallels the growth of the size of the Internet during that time. Figure5.1 shows the growth of the Internet and the corresponding growth of reported security incidents.

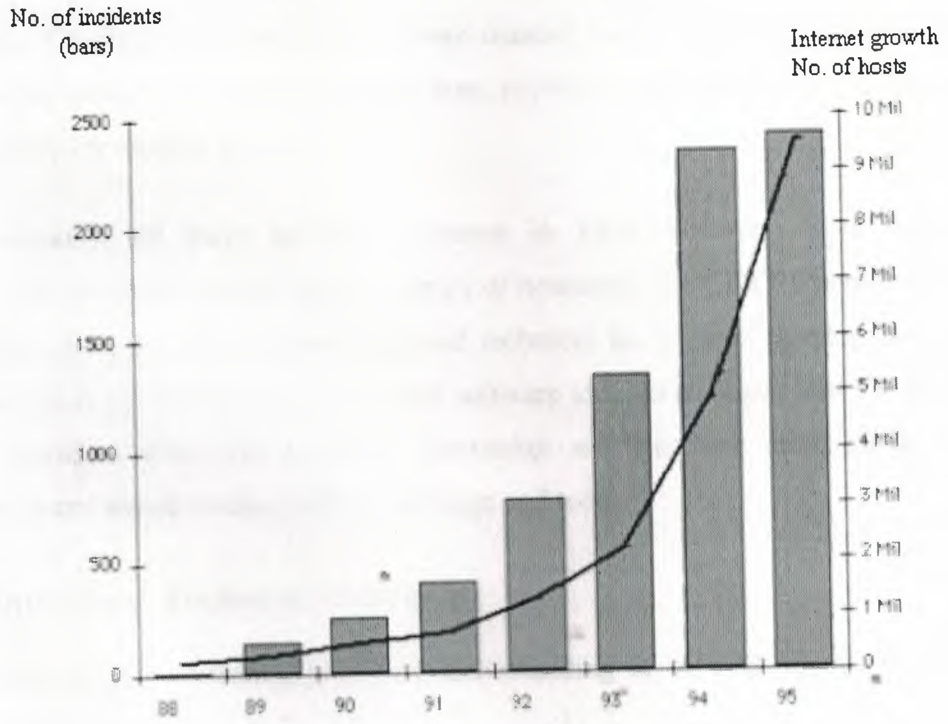


Figure5.1 Growth in Security Incidents

The data for 1995 and partial data for 1996 show a slowing of the rate at which incidents are reported to the CERT/CC (perhaps because of sites' increased security efforts or the significant increase in other response teams formed to handle incidents). However, the rate continues to increase for serious incidents, such as root compromises, services outages, and packet sniffers.

5.7 Incident Trends

In the late 1980s and early 1990s, the typical intrusion was fairly straightforward. Intruders most often exploited relatively simple weaknesses, such as poor passwords and misconfigured systems, that allowed greater access to the system than was intended. Once on a system, the intruders exploited one or another well-known, but usually unfixed, vulnerability to gain privileged access, enabling them to use the system as they wished.

There was little need to be more sophisticated because these simple techniques were effective. Vendors delivered systems with default settings that made it easy to break into systems. Configuring systems in a secure manner was not straightforward, and many system administrators did not have the time, expertise, or tools to monitor their systems adequately for intruder activity.

Unfortunately, all these activities continue in 1996; however, more sophisticated intrusions are now common. In eight years of operation, the CERT Coordination Center has seen intruders demonstrate increased technical knowledge, develop new ways to exploit system vulnerabilities, and create software tools to automate attacks. At the same time, intruders with little technical knowledge are becoming more effective as the sophisticated intruders share their knowledge and tools.

5.8 Intruders' Technical knowledge

Intruders are demonstrating increased understanding of network topology, operations, and protocols, resulting in the infrastructure attacks described in the previous section on Internet infrastructure attacks. Instead of simply exploiting well-known vulnerabilities, intruders examine source code to discover weaknesses in certain programs, such as those used for electronic mail. Much source code is easy to obtain from programmers who make their work freely available on the Internet. Programs written for research purposes (with little thought for security) or written by naive programmers become widely used, with source code available to all. Moreover, the targets of many computer intrusions are organizations that maintain copies of proprietary source code (often the source code to computer operating systems or key software utilities). Once intruders gain access, they can examine this code to discover weaknesses.

Intruders keep up with new technology. For example, intruders now exploit vulnerabilities associated with the World Wide Web to gain unauthorized access to systems. Other aspects of the new sophistication of intruders include the targeting of the network infrastructure (such as network routers and firewalls) and the ability to cloak their behavior. Intruders use Trojan horses to hide their activity from network administrators; for example, intruders alter authentication and logging programs so that they can log in without the activity showing up in the system logs. Intruders also encrypt output from their activity, such as the information captured by packet sniffers. Even if the victim finds the sniffer logs, it is difficult or impossible to determine what information was compromised.

5.9 Techniques to Exploit Vulnerabilities

As intruders become more sophisticated, they identify new and increasingly complex methods of attack. For example, intruders are developing sophisticated techniques to monitor the Internet for new connections. Newly connected systems are often not fully configured from a security perspective and are, therefore, vulnerable to attacks.

The most widely publicized of the newer types of intrusion is the use of the packet sniffers described in the section above on packet sniffers. Other tools are used to construct packets with forged addresses; one use of these tools is to mount a denial-of-service attack in a way that obscures the source of the attack. Intruders also "spoof" computer addresses, masking their real identity and successfully making connections that would not otherwise be permitted. In this way, they exploit trust relationships between computers.

With their sophisticated technical knowledge and understanding of the network, intruders are increasingly exploiting network interconnections. They move through the Internet infrastructure, attacking areas on which many people and systems depend. Infrastructure attacks are even more threatening because legitimate network managers and administrators typically think about protecting systems and parts of the infrastructure rather than the infrastructure as a whole.

In the first quarter of 1996, 7.5% of 346 incidents handled by the CERT Coordination Center involved these new and sophisticated methods, including packet sniffers, spoofing, and infrastructure attacks. A full 20% involved the total compromise of systems, in which intruders gain system-level, or root, privileges. This represents a significant increase in such attacks over previous years' attacks, and the numbers are still rising. Of 341 incidents in the third quarter of 1996, nearly 9% involved sophisticated attacks, and root compromises accounted for 33%.

5.10 Intruders' Use of Software Tools

The tools available to launch an attack have become more effective, easier to use, and more accessible to people without an in-depth knowledge of computer systems. Often a sophisticated intruder embeds an attack procedure in a program and widely distributes it to the intruder community. Thus, people who have the desire but not the technical skill are able to break into systems. Indeed, there have been instances of intruders breaking into a UNIX system using a relatively sophisticated attack and then attempting to run DOS commands (commands that apply to an entirely different operating system).

Tools are available to examine programs for vulnerabilities even in the absence of source code. Though these tools can help system administrators identify problems, they also help intruders find new ways to break into systems.

As in many areas of computing, the tools used by intruders have become more automated, allowing intruders to gather information about thousands of Internet hosts quickly and with minimum effort. These tools can scan entire networks from a remote location and identify individual hosts with specific weaknesses. Intruders may catalog the information for later exploitation, share or trade with other intruders, or attack immediately. The increased availability and usability of scanning tools means that even technically naive, would-be intruders can find new sites and particular vulnerabilities.

Some tools automate multiphase attacks in which several small components are combined to achieve a particular end. For example, intruders can use a tool to mount a denial-of-service attack on a machine and spoof that machine's address to subvert the intended victim's machine. A second example is using a packet sniffer to get router or firewall passwords, logging in to the firewall to disable filters, then using a network file service to read data on an otherwise secure server.

The trend toward automation can be seen in the distribution of software packages containing a variety of tools to exploit vulnerabilities. These packages are often maintained by competent programmers and are distributed complete with version numbers and documentation.

A typical tool package might include the following:

- network scanner
- password cracking tool and large dictionaries
- packet sniffer
- variety of Trojan horse programs and libraries
- tools for selectively modifying system log files
- tools to conceal current activity
- tools for automatically modifying system configuration files
- tools for reporting bogus checksums

5.11 Internet Vulnerabilities

A vulnerability is a weakness that a person can exploit to accomplish something that is not authorized or intended as legitimate use of a network or system. When a vulnerability is exploited to compromise the security of systems or information on those systems, the result is a security incident. Vulnerabilities may be caused by engineering or design errors, or faulty implementation.

5.11.1 Why the Internet Is Vulnerable

Many early network protocols that now form part of the Internet infrastructure were designed without security in mind. Without a fundamentally secure infrastructure, network defense becomes more difficult. Furthermore, the Internet is an extremely dynamic environment, in terms of both topology and emerging technology.

Because of the inherent openness of the Internet and the original design of the protocols, Internet attacks in general are quick, easy, inexpensive, and may be hard to detect or trace. An attacker does not have to be physically present to carry out the attack. In fact, many attacks can be launched readily from anywhere in the world - and the location of the attacker can easily be hidden. Nor is it always necessary to "break in" to a site (gain privileges on it) to compromise confidentiality, integrity, or availability of its information or service.

Even so, many sites place unwarranted trust in the Internet. It is common for sites to be unaware of the risks or unconcerned about the amount of trust they place in the Internet. They may not be aware of what can happen to their information and systems. They may believe that their site will not be a target or that precautions they have taken are sufficient. Because the technology is constantly changing and intruders are constantly developing new tools and techniques, solutions do not remain effective indefinitely. Since much of the traffic on the Internet is not encrypted, confidentiality and integrity are difficult to achieve. This situation undermines not only applications (such as financial applications that are network-based) but also more fundamental mechanisms such as authentication and nonrepudiation (see the section on basic security concepts for definitions). As a result, sites may be affected by a security compromise at another site over which they have no control. An example of this is a packet sniffer that is installed at one site but allows the intruder to gather information about other domains (possibly in other countries).

Another factor that contributes to the vulnerability of the Internet is the rapid growth and use of the network, accompanied by rapid deployment of network services involving complex applications. Often, these services are not designed, configured, or maintained securely. In the rush to get new products to market, developers do not adequately ensure that they do not repeat previous mistakes or introduce new vulnerabilities. Compounding the problem, operating system security is rarely a purchase criterion. Commercial operating system vendors often report that sales are driven by customer demand for performance, price, ease of use, maintenance, and support. As a result, off-the-shelf operating systems are shipped in an easy-to-use but insecure configuration that allows sites to use the system soon after installation. These hosts/sites are often not fully configured from a security perspective before connecting. This lack of secure configuration makes them vulnerable to attacks, which sometimes occur within minutes of connection.

Finally, the explosive growth of the Internet has expanded the need for well-trained and experienced people to engineer and manage the network in a secure manner. Because the need for network security experts far exceeds the supply, inexperienced people are called upon to secure systems, opening windows of opportunity for the intruder community.

5.11.2 Types of Technical Vulnerabilities

The following taxonomy is useful in understanding the technical causes behind successful intrusion techniques, and helps experts identify general solutions for addressing each type of problem.

5.12 Flaws in Software or protocol Designs

Protocols define the rules and conventions for computers to communicate on a network. If a protocol has a fundamental design flaw, it is vulnerable to exploitation no matter how well it is implemented. An example of this is the Network File System (NFS), which allows systems to share files. This protocol does not include a provision for authentication; that is, there is no way of verifying that a person logging in really is whom he or she claims to be. NFS servers are targets for the intruder community.

When software is designed or specified, often security is left out of the initial description and is later "added on" to the system. Because the additional components were not part of the original design, the software may not behave as planned and unexpected vulnerabilities may be present.

5.12.1 Weaknesses in How Protocols and Software Are Implemented

Even when a protocol is well designed, it can be vulnerable because of the way it is implemented. For example, a protocol for electronic mail may be implemented in a way that permits intruders to connect to the mail port of the victim's machine and fool the machine into performing a task not intended by the service. If intruders supply certain data for the "To:" field instead of a correct E-mail address, they may be able to fool the machine into sending them user and password information or granting them access to the victim's machine with privileges to read protected files or run programs on the system. This type of vulnerability enables intruders to attack the victim's machine from remote sites without access to an account on the victim's system. This type of attack often is just a first step, leading to the exploitation of flaws in system or application software.

Software may be vulnerable because of flaws that were not identified before the software was released. This type of vulnerability has a wide range of subclasses, which intruders often exploit using their own attack tools. For readers who are familiar with software design, the following examples of subclasses are included:

- race conditions in file access
- non-existent checking of data content and size
- non-existent checking for success or failure
- inability to adapt to resource exhaustion
- incomplete checking of operating environment
- inappropriate use of system calls
- re-use of software modules for purposes other than their intended ones

By exploiting program weaknesses, intruders at a remote site can gain access to a victim's system. Even if they have access to a nonprivileged user account on the victim's system, they can often gain additional, unauthorized privileges.

5.12.2 Weaknesses in System and Network Configuration

Vulnerabilities in the category of system and network configurations are not caused by problems inherent in protocols or software programs. Rather, the vulnerabilities are a result of the way these components are set up and used. Products may be delivered with default settings that intruders can exploit. System administrators and users may neglect to change the default settings, or they may simply set up their system to operate in a way that leaves the network vulnerable.

An example of a faulty configuration that has been exploited is anonymous File Transfer Protocol (FTP) service. Secure configuration guidelines for this service stress the need to ensure that the password file, archive tree, and ancillary software are separate from the rest of the operating system, and that the operating system cannot be reached from this staging area. When sites misconfigure their anonymous FTP archives, unauthorized users can get authentication information and use it to compromise the system.

5.13 Security Policy, Procedures, and Practices

5.13.1 Security Policy

A policy is a documented high-level plan for organization-wide computer and information security. It provides a framework for making specific decisions, such as which defense mechanisms to use and how to configure services, and is the basis for developing secure programming guidelines and procedures for users and system administrators to follow. Because a security policy is a long-term document, the contents avoid technology-specific issues. A security policy covers the following (among other topics appropriate to the organization):

- high-level description of the technical environment of the site, the legal environment (governing laws), the authority of the policy, and the basic philosophy to be used when interpreting the policy
- risk analysis that identifies the site's assets, the threats that exist against those assets, and the costs of asset loss
- guidelines for system administrators on how to manage systems
- definition of acceptable use for users
- guidelines for reacting to a site compromise (e.g., how to deal with the media and law enforcement, and whether to trace the intruder or shutdown and rebuild the system)

Factors that contribute to the success of a security policy include management commitment, technological support for enforcing the policy, effective dissemination of the policy, and the security awareness of all users. Management assigns responsibility for security, provides training for security personnel, and allocates funds to security. Technological support for the security policy moves some responsibility for enforcement from individuals to technology.

The result is an automatic and consistent enforcement of policies, such as those for access and authentication. Technical options that support policy include (but are not limited to)

- challenge/response systems for authentication
- auditing systems for accountability and event reconstruction
- encryption systems for the confidential storage and transmission of data
- network tools such as firewalls and proxy servers

5.13.2 Security-Related Procedures

Procedures are specific steps to follow that are based on the computer security policy. Procedures address such topics as retrieving programs from the network, connecting to the site's system from home or while traveling, using encryption, authentication for issuing accounts, configuration, and monitoring.

5.13.3 Security Practices

System administration practices play a key role in network security. Checklists and general advice on good security practices are readily available. Below are examples of commonly recommended practices:

- Ensure all accounts have a password and that the passwords are difficult to guess. A one-time password system is preferable.
- Use tools such as MD5 checksums (8), a strong cryptographic technique, to ensure the integrity of system software on a regular basis.
- Use secure programming techniques when writing software. These can be found at security-related sites on the World Wide Web.
- Be vigilant in network use and configuration, making changes as vulnerabilities become known.
- Regularly check with vendors for the latest available fixes and keep systems current with upgrades and patches.
- Regularly check on-line security archives, such as those maintained by incident response teams, for security alerts and technical advice.

- Audit systems and networks, and regularly check logs. Many sites that suffer computer security incidents report that insufficient audit data is collected, so detecting and tracing an intrusion is difficult.

5.13.4 Security Technology

A variety of technologies have been developed to help organizations secure their systems and information against intruders. These technologies help protect systems and information against attacks, detect unusual or suspicious activities, and respond to events that affect security. In this section, the focus is on two core areas: operational technology and cryptography. The purpose of operational technology is to maintain and defend the availability of data resources in a secure manner. The purpose of cryptography is to secure the confidentiality, integrity, and authenticity of data resources.

5.14 Operational Technology

Intruders actively seek ways to access networks and hosts. Armed with knowledge about specific vulnerabilities, social engineering techniques, and tools to automate information gathering and systems infiltration, intruders can often gain entry into systems with disconcerting ease. System administrators face the dilemma of maximizing the availability of system services to valid users while minimizing the susceptibility of complex network infrastructures to attack. Unfortunately, services often depend on the same characteristics of systems and network protocols that make them susceptible to compromise by intruders. In response, technologies have evolved to reduce the impact of such threats. No single technology addresses all the problems. Nevertheless, organizations can significantly improve their resistance to attack by carefully preparing and strategically deploying personnel and operational technologies. Data resources and assets can be protected, suspicious activity can be detected and assessed, and appropriate responses can be made to security events as they occur.

One-Time Passwords Intruders often install packet sniffers to capture passwords as they traverse networks during remote log-in processes. Therefore, all passwords should at least be encrypted as they traverse networks. A better solution is to use one-time passwords because there are times when a password is required to initiate a connection before confidentiality can be protected.

One common example occurs in remote dial-up connections. Remote users, such as those traveling on business, dial in to their organization's modem pool to access network and data resources. To identify and authenticate themselves to the dial-up server, they must enter a user ID and password. Because this initial exchange between the user and server may be monitored by intruders, it is essential that the passwords are not reusable. In other words, intruders should not be able to gain access by masquerading as a legitimate user using a password they have captured.

One-time password technologies address this problem. Remote users carry a device synchronized with software and hardware on the dial-up server. The device displays random passwords, each of which remains in effect for a limited time period (typically 60 seconds). These passwords are never repeated and are valid only for a specific user during the period that each is displayed. In addition, users are often limited to one successful use of any given password. One-time password technologies significantly reduce unauthorized entry at gateways requiring an initial password.

Monitoring Tools Continuous monitoring of network activity is required if a site is to maintain confidence in the security of its network and data resources. Network monitors may be installed at strategic locations to collect and examine information continuously that may indicate suspicious activity. It is possible to have automatic notifications alert system administrators when the monitor detects anomalous readings, such as a burst of activity that may indicate a denial-of-service attempt. Such notifications may use a variety of channels, including electronic mail and mobile paging. Sophisticated systems capable of reacting to questionable network activity may be implemented to disconnect and block suspect connections, limit or disable affected services, isolate affected systems, and collect evidence for subsequent analysis.

Tools to scan, monitor, and eradicate viruses can identify and destroy malicious programs that may have inadvertently been transmitted onto host systems. The damage potential of viruses ranges from mere annoyance (e.g., an unexpected "Happy Holidays" jingle without further effect) to the obliteration of critical data resources. To ensure continued protection, the virus identification data on which such tools depend must be kept up to date. Most virus tool vendors provide subscription services or other distribution facilities to help customers keep up to date with the latest viral strains.

Security Analysis Tools Because of the increasing sophistication of intruder methods and the vulnerabilities present in commonly used applications, it is essential to assess periodically network susceptibility to compromise. A variety of vulnerability identification tools are available, which have garnered both praise and criticism. System administrators find these tools useful in identifying weaknesses in their systems. Critics argue that such tools, especially those freely available to the Internet community, pose a threat if acquired and misused by intruders.

5.15 Information Warfare

Extensive and widespread dependence on the Internet has called new attention to the importance of information to national security. The term information warfare refers to the act of war against the information resources of an adversary. Like warfare on land or in the air, information warfare is one component of a range of attack strategies for dominating an adversary in order to gain or maintain an objective.

Information warfare is divided into two categories: offensive and defensive. The purpose of offensive information warfare is to attack the information resources of an adversary to gain dominance. Defensive information warfare is the protection of your information assets against attack.

Information assets can take many forms, from messages sent by courier in diplomatic bags to the computers used to analyze enemy positions based on satellite data. In computer security, information assets include digital information, the computers that process them, and the networks that transmit the digital information from place to place. Computer security is a key element for protecting the availability, integrity, and confidentiality of all these information assets.

Internet security protects information assets consisting of computers, information, and networks that are part of the Internet. Internet security is related to information warfare when the Internet contains information assets that are important to the information warfare objective. For example, if an adversary can use the Internet to access battle plans, the Internet is being used for information warfare. Internet security is important to both offensive and defensive information warfare because the Internet is a global and dependable resource on which many countries rely. Historically, military networks and computers were unreachable by nonmilitary participants. The Internet, however, provides a cost-effective way for military and government units to communicate and participate in achieving objectives. Use of the Internet means that individuals, multinational companies, and terrorist organizations all can gain access to important information resources of governments and military forces. Thus, it is important to address Internet security concerns as a key component of defensive information warfare. Because the Internet is global, it can be an avenue of attack for offensive information warfare by many governments. One of the battlefields for a future military offensive could very well involve the Internet. Intruder technology (as described in a separate section above) could be used by a government as a weapon against information resources, or used randomly by a terrorist organization against civilian targets. In the study of information warfare, there are many new problems to solve that are not evident in other forms of warfare. These problems include identifying the enemy, responding without making your systems vulnerable to attack, and gathering intelligence on the Internet about preparations for a military exercise. These and other problems are likely to be the subject of discussion and investigation for some time to come.

5.16 Summary

In this chapter we discussed the internet security, basic security concepts, why care about security, network security incidents, sources of incidents, types of incidents, the vulnerabilities of internet, flaws in software or protocol design, security policy, procedures, practices and last the operational technology of security.

CONCLUSIONS

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

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

REFERNCES

[1] Tanebaum Andrew S., Computer Networks, 1996

[2] Martin Michael J., Understanding the Network: A Practical Guide to Internetworking, Macmillan Computer publishing,USA, 2000

[3] Microsoft, Networking Essentials, Microsoft Corporation, Washington, 1996

[4] Draft Standard IEEE 802.11. Wireless LAN Medium Access Control(MAC) and Physical Layer (PHY) Specifications. IEEE, 1996

[5] K. Pahlavan, A.Zahedi, P. Krishnamurthy. Evolving WLAN Industry Product and Standards. Invited paper PIMRC'97, Worcester Polytechnic Institute, 1997

[6] A.Zahedi, P. Krishnamurthy, S. Bagchi, K. Pahlavan. An update on the Evolution of the Wireless LAN services. Worcester Polytechnic Institute, 1997