



Implementing a Secure Windows 2000 Servers for the Internet

by

Mr. Charoenchai Tulyagijja

A Final Report of the Six-Credit Course
IC 6998 E-Commerce Practicum

Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science
in Internet and E-Commerce Technology
Assumption University

July 2003

Implementing a Secure Windows 2000 Servers for the Internet

by
Mr. Charoenchai Tulyagijja

A Final Report of the Six-Credit Course
IC 6998 E-Commerce Practicum


Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science
in Internet and E-Commerce Technology
Assumption University

July 2003


Project Title Implementing a Secure Windows 2000 Service for the Internet
Name Mr. Charoenchai Tulyagijja
Project Advisor Rear Admiral Prasart Sribhadung
Academic Year July 2003


The Graduate School of Assumption University has approved this final report of the three-credit course, IC 6997 and IC 6999 E-Commerce Practicum, submitted in partial fulfillment of the requirements for the degree of Master of Science in Internet and E-Commerce Technology.

Approval Committee:


(Rear Admiral Prasart Sribhadung)
Dean and Advisor


(Prof. Dr. Srisakdi Charmonman)
Chairman


(Dr. Ketchayong Skowratananont)
Member


(Assoc. Prof. Somchai Thayarnyong)
MUA Representative

July 2003

ABSTRACT

This project identifies internet security issues, comparing the security features between Windows NT and 2000 and demonstrating the perimeter network concept. Moreover, this project also includes the techniques on how to install, configure, set up remote administration, backup, restore and audit a secure Windows 2000 server.

This project concentrates only on necessary information which can make a Windows 2000 server more reliable and can be trusted for critical business application.



ACKNOWLEDGEMENTS

The writer is indebted to the following people for without them, this project would not have been possible.

The writer wishes to express sincere gratitude to his advisor Rear Admiral Prasart Sribhadung. The advisor's patience, guidance and constant encouragement has led the writer to this completion. The writer would like to take this opportunity to thank the entire faculty that taught him during his Master of Science in Internet and E-commerce Technology. The knowledge that he acquired from them indeed helped make this a successful report.

The writer would also like to thank his friends, Mr. Pradt Kovitariyavongse. He always asked about the progress of this paper. Thanks for his advice, ideas and encouragement and everyone who is concerned.

Finally, the writer would like to thank his parents, sister and friends for their support and patience throughout the project.

TABLE OF CONTENTS

<u>Chapter</u>	<u>Page</u>
ABSTRACT	i
ACKNOWLEDGEMENTS	ii
LIST OF FIGURES	v
LIST OF TABLES	vii
I. INTRODUCTION	1
1.1 Background	1
1.2 Objectives	1
1.3 Scope	2
1.4 Research Methodology	2
1.5 Deliverable	2
II. WINDOWS 2000 SECURITY	3
2.1 Problem Formulation	3
2.2 Building a Secure Site on the Internet	7
2.3 The Windows 2000 Architectures	21
2.4 Windows 2000 in the Perimeter Network	29
2.5 Cryptography Basics	35
III. WINDOWS 2000 VERSUS WINDOWS NT 4.0	40
3.1 Security Templates and Security Configuration Analysis	40
3.2 Group Policy	47
3.3 IPSec	48
3.4 Kerberos	51
3.5 Encrypting File System (EFS)	53
3.6 Runas	54

<u>Chapter</u>	<u>Page</u>
3.7 A Windows File Protection (WFP)	56
3.8 Summary	57
IV. SECURING WINDOWS 2000 SERVERS	60
4.1 Installation and Configuration	60
4.2 Setting up Secure Remote Administration	101
4.3 Backing up and Restoring	110
4.4 Auditing and Monitoring	117
4.5 Summary	129
V. CONCLUSIONS AND RECOMMENDATIONS	131
5.1 Conclusions	131
5.2 Recommendations	131
BIBLIOGRAPHY	133



LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
2.1 Number of Incidents Reported to CERT-CC	6
2.2 An Enterprise with Three Perimeter Networks	12
2.3 Security Zones in the Perimeter	13
2.4 A Perimeter with One Security Zone	13
2.5 Perimeter with Two Security Zones	14
2.6 An Application-level Gateway	16
2.7 A Packet-filtering Device	17
2.8 Example Perimeter Network	19
2.9 The Windows NT Architecture	22
2.10 The Windows NT Networking Architecture	26
2.11 Example of Bindings in the Windows Network Architecture	29
2.12 Dual Firewall Systems and Reverse Proxy Solution	32
2.13 NT Domains in the Perimeter	34
3.1 An MMC Window with the Security Configuration and Analysis and Security Template Snap-ins Installed	41
3.2 The Results of a Security Analysis	46
3.3 The Kerbtray Utility Show Kerberos Tickets Held by the Local Machine	52
4.1 The Windows 2000 Configuration and Bindings for a Network Card	61
4.2 A List of the Services Running on My Test System	64
4.3 The Windows 2000 Device Manager	65
4.4 The Security-related Processes and Components	68
4.5 The Syskey Command Encryption Dialog Box	70
4.6 The Syskey System Password Option Dialog Box	71

<u>Figure</u>	<u>Page</u>
4.7 The TCP/IP Security Filters Can Be Used to Block Inbound Traffic	82
4.8 The TCP/IP Filtering Dialog Box in Windows 2000	83
4.9 The TCP Three-way Handshake	84
4.10 The Windows Components Dialog Box	103
4.11 The Terminal Services Mode of Operation Configuration	104
4.12 The Terminal Services Configuration MMC Snap-in	105
4.13 RDP Encryption Settings	106
4.14 RDP Sessions Settings Tab	108
4.15 The RDP Permissions Tab	109
4.16 A Common Network Backup Design in Perimeter Networks	112
4.17 An Example of a Poor Backup Network Design	113
4.18 An Example of a Good Backup Network Design	113
4.19 Windows Backup (Windows 2000)	117
4.20 Additional Registry Setting	123
4.21 The ACL Editor in Windows 2000	127
4.22 Configuring Auditing Using the Registry Editor	128
4.23 The Recommended Auditing Settings on HKLM\System	129

LIST OF TABLES

<u>Table</u>	<u>Page</u>
3.1 Definitions of the Default Windows 2000 Security Templates	43
4.1 Windows 2000 User Mode Processes Running after Disabling Unnecessary	65
4.2 Recommended Account Policy	74
4.3 Windows 2000 Rights and Privileges	76
4.4 The SynAttack Setting	86
4.5 The SYN/ACK Setting	87
4.6 The Dynamic Backlog Setting	89
4.7 The TCP Keep-alive Timer	90
4.8 The MTU Discovery Setting	91
4.9 The Source Routing Setting	92
4.10 The Dead Gateway Detection Setting	92
4.11 The Router Discovery Setting	93
4.12 The ICMP Redirects Setting	93
4.13 The RPC Setting	95
4.14 The Administrative Tool and Utility files	96
4.15 The NTFS Permission	98
4.16 The Predefined Groups' Permission Setting for File Access	99
4.17 The Predefined Groups' Permission Setting for Directory Access	99
4.18 The Registry Permissions	100
4.19 Recommended Session Settings for Terminal Services	107
4.20 Log Files Setting	120
4.21 Security Audit Failure Setting	121
4.22 Additional Registry Setting	122



I. INTRODUCTION

1.1 Introduction

Until recently, few system administrators believed that Windows NT was a reasonable platform to use as a web server or as any other type of system exposed to the Internet. Most Internet servers have historically been based on Unix, and most experienced system administrators have regarded Windows NT as insecure, suitable for use as a file or print server, but not to be trusted for critical business applications.

Unix still tends to be the first choice as a platform for an Internet server, but as Windows NT becomes more secure, and as more administrators gain experience with it, Windows NT (and now Windows 2000) systems are emerging as viable platforms for Internet servers. More and more organizations are now entrusting the full Spectrum of business activities to Windows NT. Today, about 20% of all web servers on the Internet are using Windows NT, many of them for e-commerce.

This project is designed with the purpose of providing all necessary information on how to implement secure Windows 2000 Servers for the Internet.

1.2 Objective

- (1) To conduct a research on the security of Windows 2000, identify internet security issues, and the perimeter network.
- (2) To uncover the Windows 2000 Internet security issues.
- (3) To demonstrate how to harden a Windows 2000 server including secure remote administration.
- (4) To show how to back up, restore, audit and monitor the Windows 2000 Server.

1.3. Scope

- (1) Research about Windows 2000 security threats, security architecture of the Windows 2000 operating systems, and typical perimeter network configurations.
- (2) Research how to harden the Windows 2000 bastion hosts: configuring Services, editing the registry, and setting permissions.
- (3) Secure remote administration using Windows 2000 Terminal.
- (4) Windows 2000 backup, recovery, auditing and monitoring.

1.4. Research Methodology

This research is using documentary and investigative research.

1.5. Deliverable

The final report covers the scope as afore mentioned.



II. WINDOWS 2000 SECURITY

The use of Windows systems as Internet servers presents security challenges. In contrast to most internal systems, systems connected to the Internet are directly exposed to security attacks from both unsophisticated and highly skilled attackers. The typical Windows 2000 installation makes a Windows server an easy target for such attacks. Securing the Windows 2000 operating system for Internet use is a complex task. The purpose of this paper is to offer a strategy for making your Windows-based server configuration as secure as possible. This strategy has two basic parts:

- (1) Secure or “harden” any Windows server that will be exposed to potential attacks from the Internet so it is as secure as it possibly can be. An exposed system of this kind is typically known as a bastion host.
- (2) Provide extra security protection for such exposed systems by installing an additional network—typically known as a perimeter network—that separates the outside network (usually the Internet) from your organization’s internal networks.

This paper describes specifically how to harden your Windows 2000 system so it can function on your perimeter network as a secure bastion host. Before I present the step-by-step security details, this chapter sets the scene by describing briefly the security threats your system will face, the architecture of Windows 2000 operating systems, and the recommended placement of Windows servers on your perimeter network.

2.1 Problem Formulation

An Internet server faces many different kinds of threats. The most common include:

Intrusion

An intrusion occurs when an unauthorized person gains access to a system. These days, intrusions most often result in web page defacement: an attacker alters the contents of the web site. Such attacks are growing in popularity. Attrition (<http://www.attrition.org/mirror/attrition/>) maintains a daily updated list of defaced web sites. The current record is 56 reported defacements in one day (November 21, 1999). About 60% of the defacements recorded at Attrition between October 1999 and April 2000 have occurred on Windows NT systems.

Denial of Service

The goal of a denial of service (DOS) attack is to sabotage operation by consuming all of your computing resources (CPU time, network bandwidth, etc.). This effectively stops authorized users from using the system.

Information Theft

This type of attack occurs when an unauthorized person obtains private information. The most popular targets are login/password information, credit card information, and software source code.

Many intrusions are made possible by improperly configured software. Looking at a concrete example may help underscore this point. Recently, the Apache web server site (<http://www.apache.org/>) was hacked. In this particular case, the attackers uploaded a PHP script to a world-writeable FTP directory. The web server root directory was the same as the FTP server root directory. This allowed the attackers to launch Unix commands using the uploaded script. They uploaded and executed a shell binary that bound to a high port and enabled them to initiate a connection to that port. The attackers now had interactive shell access on the system. The next step was to gain root access.

This was accomplished by using a database process that was running as root to indirectly create a setuid root shell.

Fortunately, these attackers (so-called “gray-hats”) were not out to thrash the site; they only replaced the “powered by Apache” logo with a Microsoft Back Office logo and alerted the site administrators.

The following configuration errors made the Apache break-in possible:

- (1) The web server and the FTP server had the same root directory. This allowed the attackers to upload the software that was used to launch the attack. The uploaded software could be executed because the web server software used the same filesystem hierarchy.
- (2) There was no (or an improperly configured) firewall system protecting the web server. It was possible for the attackers to connect to any port on the system. This made the attack much easier.
- (3) The database software was running as root. This is the reason why the attackers were eventually able to gain root access.

Windows NT systems present many vulnerabilities, which attackers are only too happy to take advantage of. For example, there are cases where an attacker has been able to connect directly to a system using Windows file sharing. Those systems are an even easier target than the Apache site was. Just start guessing passwords and try to connect as Administrator!

The number of security incidents reported to the Computer Emergency Response Team Coordination Center (CERT-CC) has grown at an alarming rate in recent years. Figure 2.1 illustrates this development; note how steeply incidents have increased since 1997. (Incidents include, but are not limited to, attempts to gain unauthorized access to a system or its data, and disruption or denial of service.) The real security picture is far

worse than these statistics show; it is safe to assume that only a small number of all incidents are reported to CERT-CC.

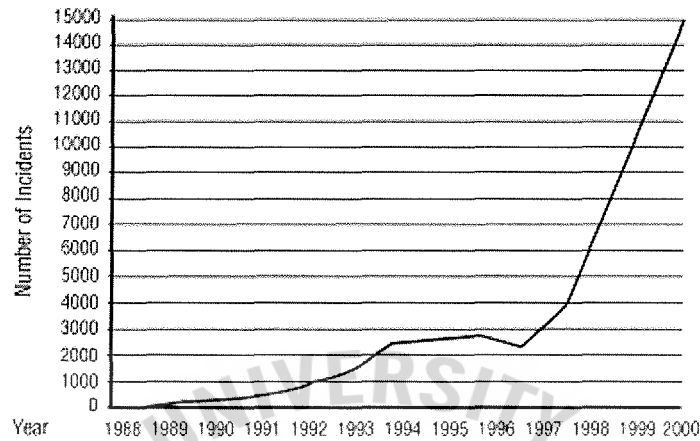


Figure 2.1. Number of Incidents Reported to CERT-CC.

If you already have a presence on the Internet, you probably know that attempts are made to compromise your site's security mechanisms every day. And the stakes are high. Imagine how you'd feel if an online store that you use was hacked and the attacker managed to steal your credit card information. Would you feel comfortable shopping there again? Would you use an Internet bank that was successfully attacked last year? I wouldn't.

As a result, it is of great importance to have and maintain a high level of security for your site. This is a complex task, but after reading this paper I hope you will find it somewhat less troublesome.

Keep in mind that even if you're not running a large online bank or shopping site, you still need to take steps to protect your servers. The attackers are out there; they may be after your intellectual property; they may want to use your computing resources; or they may just want to have some fun defacing your web site.

2.2 Building a Secure Site on the Internet

Building and maintaining a secure site on the Internet includes many more tasks than simply installing your operating system, however securely you may do so. Overall security is a combination of secure software and careful human planning and administration. You will need to be concerned with all of the following tasks:

Planning

Securing an Internet site must be a carefully planned and coordinated process. It is not just a matter of clicking on screens and working it out as you go. Figure out the goals and tactics ahead of time, and then implement security, step-by-step. It is also important to understand that you need one encompassing plan that includes all aspects of the process, rather than several small and uncoordinated planning efforts.

Policies

In order to achieve a high level of security, you need policies that define the main aspects of running an Internet site. This is not a paper on policies, but keep in mind that before you start building a secure system, you need to have the appropriate policies in place. Start by reading the Site Security Handbook (RFC 2196); it is an excellent introduction to this topic.

Access Control

Access control protects systems from unauthorized use; there are several different types:

Physical Access Control

Physical access control is often overlooked, but it is an extremely important outer level of protection. Large organizations often have big computer rooms that are both bomb-proof and earthquake-proof, which is good. In many cases, however, pretty much

everyone in the organization has access to these rooms, which makes it possible for anyone in the building to sabotage operations.

System Access Control

Only the people involved in the daily operation of your systems should have access to these systems. Those who are granted access should have only the amount of privilege required to do their jobs. For example, not everyone needs to be a member of the Administrator group.

Network Access Control

Network access to your systems needs to be restricted by a firewall system. A firewall system consists of a number of components that act in concert to enforce your network access policy; it is typically not just one single gateway with firewall software installed. The perimeter network (discussed later in this chapter) is a type of firewall system.

Operation

Once your system is up and running, you need to manage its operation in a careful and secure manner. System management includes:

Auditing

Watch your systems carefully. Set up an audit policy that keeps you informed of any access policy violations. Auditing and Monitoring Your Perimeter Network deals with the different aspects of setting up auditing on a Windows NT/2000 bastion host.

Backups

Make frequent backups. Always back up before and after changing the configuration of your systems. The flip side of backup is restore; you must attempt to restore your system from backups at regular intervals to make sure you'll be able to do

so if there is a disaster. Backing Up and Restoring Your Bastion Host serves as an introduction to backing up and restoring bastion hosts.

Log Management

Collect logs in real time on a separate secured logging host and carefully review this information.

Peer Reviews

Ask your colleagues or a third party to review your work periodically.

Encryption

Use encryption to secure communication and sensitive data stored on disk. You will find references to various types of encryption methods and algorithms throughout this paper. The “Cryptography Basics” section later in this chapter provides a brief introduction.

It is important to understand that site security is a very big and complex subject and that this paper’s focus on the practical aspects of building and managing secure bastion hosts based on Windows NT/2000 is a very narrow aspect of site security.

Hardening the Bastion Host

Microsoft’s success in the network operating system market is largely because its products are very easy to use. The Windows server version has the familiar user interface that almost all office workers use every day. It is easy to get started, and you don’t need in-depth knowledge of the operating system to install a Windows NT/2000 server. Most components are configured and started automatically, just as they are in the consumer Windows 95/Windows 98 operating system. These characteristics are attractive for an internal file and print server that isn’t exposed to direct attack. However, you want something quite different for an external web server that serves the organization’s customers and partners over the Internet. A system exposed in this way

should provide a minimum of services and needs to be properly configured to ensure a higher level of security. As I mentioned earlier in this chapter, a system configured in this manner is referred to as a bastion host.

Basically, a bastion host is a computer system that is a critical component in a network security system, and one that is exposed to attack. Examples of bastion hosts are firewall gateways, web servers, FTP servers, and Domain Name Service (DNS) servers. Because bastion hosts are so important—and so vulnerable—such systems must be highly fortified. You must pay special attention to fortifying (i.e., establishing the maximum possible security for) the bastion host during both initial construction and ongoing operation.

A bastion host is a system identified by the firewall administrator as a critical strong point in the network's security. Generally, bastion hosts will have some degree of extra attention paid to their security, may undergo regular audits, and may have modified software.

Bastion hosts are not general-purpose computing resources. They differ in both their intent and their specific configuration. The process of configuring or constructing a bastion host is often referred to as hardening.

The effectiveness of a specific bastion host configuration can usually be judged by answering two questions:

- (1) How does the bastion host protect itself from attack?
- (2) How does the bastion host protect the network behind it from attack?

Exercise extreme caution when installing software on bastion hosts. Very few software products have been designed and tested to run safely on these exposed systems.

Configuring the Perimeter Network

No matter how carefully you configure your bastion host to withstand direct attacks, you can't be entirely confident about its security. Most software code has bugs in it, and therefore all systems potentially have undiscovered security vulnerabilities. For this reason, it is important to provide extra layers of security for systems that are as exposed and as vulnerable as bastion hosts.

A common way to protect exposed servers on the Internet is to implement some kind of network-based access control mechanism that serves as extra protection for the bastion hosts. One such very effective mechanism is provided by a perimeter network. A perimeter network is a network that connects your private internal network to the public Internet or another untrusted network. This makes the perimeter network very important from a security standpoint. The purpose of this network is to serve as a single point of access control. All components in a perimeter must act in concert to implement a site's firewall policy. In other words, the perimeter network is a firewall system.

The perimeter network is a key part of the architecture of many current Internet sites. The reasons are partly historical. When the Internet took off commercially, many companies wanted to get on the Net to do business. The first step was often simply to publish product information on a web server. These web servers typically contained only static information, and thus didn't need to be connected to the internal network. With the advent of e-commerce, such web servers had to be connected in some way both to the clients on the Internet and to the legacy systems on the internal network—for example, to process orders and check the availability of products.

Many companies now faced the requirement to connect their internal networks to the Internet—and to the accompanying security risks. Since the Internet could not be

trusted for obvious reasons, there was an increasing need for company-controlled networks that could act as secured perimeters.

The Perimeter Network Architecture

A perimeter network is an untrusted part of an enterprise network that resides on the outskirts of the private network. The perimeter network is often also referred to as the demilitarized zone, or DMZ, named after the region separating North Korea and South Korea. An example of a perimeter network is where the Internet connection and the web servers are located. A company might have several perimeter networks, as illustrated in Figure 2.2.

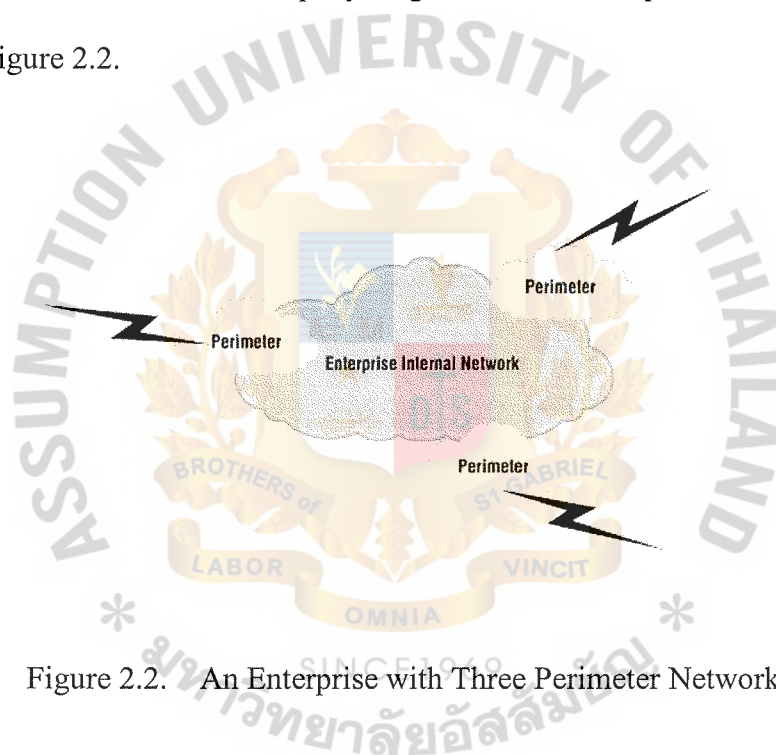


Figure 2.2. An Enterprise with Three Perimeter Networks.

All external communication from the internal network has to pass the perimeter before it can reach an external host, and no communication is allowed directly from an external network to the internal network.

A good approach to building the perimeter network is to build it in compartments, so that the perimeter is able to protect itself and the internal network even if one compartment is compromised. This compartmentalization is illustrated in Figure 2.3.

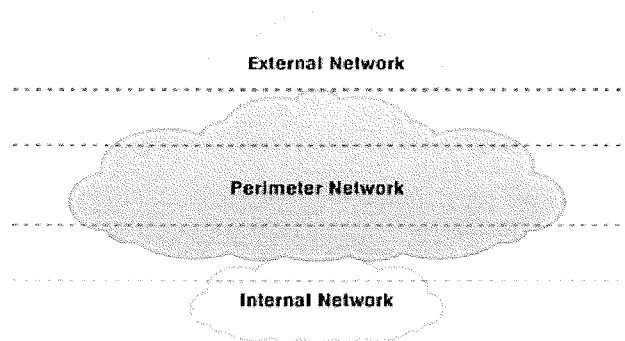


Figure 2.3. Security Zones in the Perimeter.

Since each compartment has access control mechanisms, the farther away from the external network a host is placed, the better it is protected. It is good security practice to block as much traffic as possible in each compartment layer—I recommend that you take a *default-deny* stance regarding network traffic. With a default-deny stance, everything that isn't explicitly allowed is denied, in contrast to a *default-allow* stance, where everything that isn't explicitly denied is allowed. Consider the example in Figure 2.4.

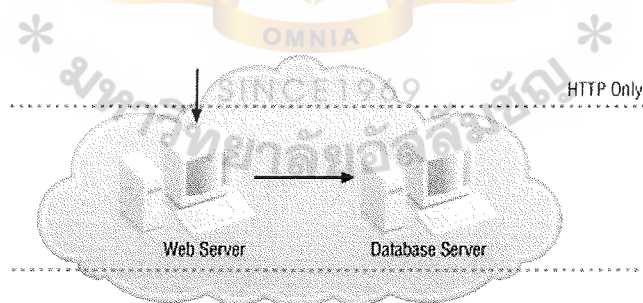


Figure 2.4. A Perimeter with One Security Zone.

In this example, if the web server is compromised, it is easy for an intruder to attack any service provided by the database server. This is because there is no network access control between these two servers.

On the other hand, in the topology shown in Figure 2.5, if the web server is compromised, the access control layer between the compartments will block unneeded traffic to the database system. As a result, the intruder may be able to attack the database process on the server, but not be able to attack anything else.

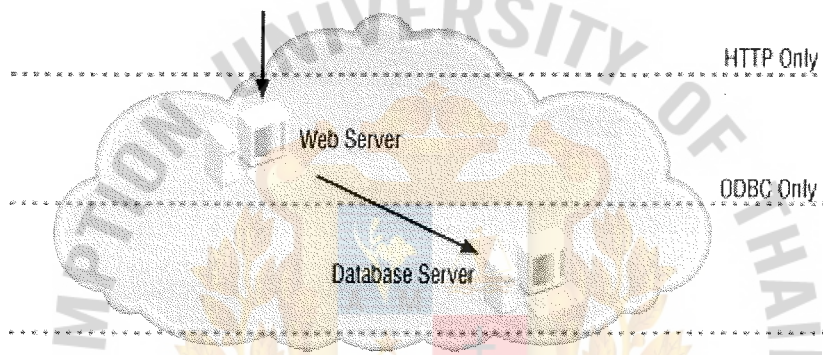


Figure 2.5. Perimeter with Two Security Zones.

Components in the Perimeter

It takes a number of different components to build a perimeter network, and some architectures are quite complex. This section does not attempt to describe all of the issues or possible combinations. It simply introduces the components and explains how they interact so you will have enough background to be able to understand subsequent chapters.

(1) Routers

Routers are the traffic police of the network. They decide what route a datagram should take at each router or “network intersection.” Like the

police, routers can also choose to stop certain types of traffic. Traffic is controlled by rules called router Access Control Lists (ACLs). Example 1-1 shows a router ACL for a Cisco router.

Example 2-1. Cisco IOS Router ACL Example

```
ip access-list extended example access_list  
permit tcp any 192.168.1.0 0.0.0.255 eq http  
permit icmp any any  
deny ip any any log
```

A Cisco IOS ACL is applied from top to bottom. An incoming datagram is tested against each line in the ACL. This example allows HTTP from anywhere to the 192.168.1/24 network. It also allows any type of ICMP anywhere. All other datagrams are blocked and logged.

Using router ACLs in this manner provides us with a useful network access control mechanism in the perimeter. A router that implements access control in this manner is generally referred to as a screening router.

(2) Firewall gateways

Certain components in the perimeter typically have firewall software installed, and these machines are referred to as firewall gateways.

There are two common techniques that a firewall gateway can use. One method, shown in Figure 2.6, is to act as an application-level gateway; the gateway serves as a middleman that intercepts traffic at the application level, and it initiates a new connection to the target system on behalf of the client. Examples of application-level protocols are File Transfer Protocol (FTP), HyperText Transfer Protocol (HTTP), Simple Mail Transfer Protocol (SMTP), and Post Office Protocol (POP).

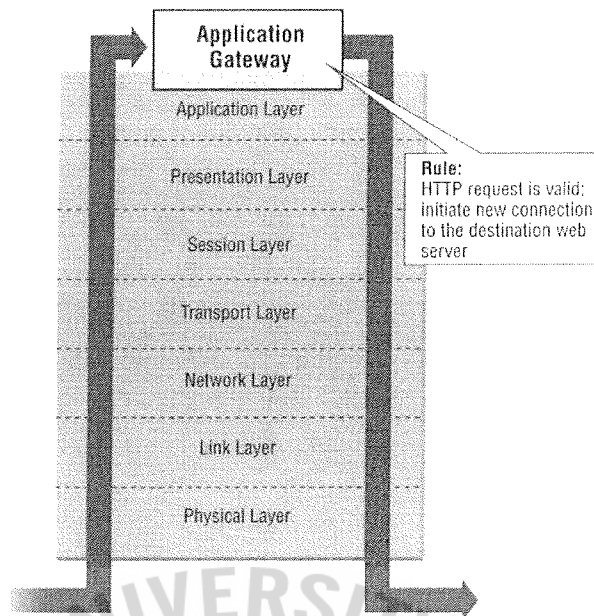


Figure 2.6. An Application-level Gateway.

The other technique, illustrated in Figure 2.7, is to inspect the traffic on the Internet Protocol (IP) level. This is called packet filtering. A more sophisticated form of packet filtering, called stateful inspection, is used by products such as Checkpoint's Firewall-1. A state-aware firewall gateway keeps track of the state of the connections that are going through it. If an outgoing HTTP request from a client is allowed through the gateway, the response to that request also has to be allowed through. The firewall software adds a temporary rule in its rulebase to allow the response from the destination web server to the client. The firewall gateway also understands some types of application data (HTTP, SMTP, FTP, etc.) in the IP datagrams, and for this reason, it may be able to make better security decisions than a screening router can.

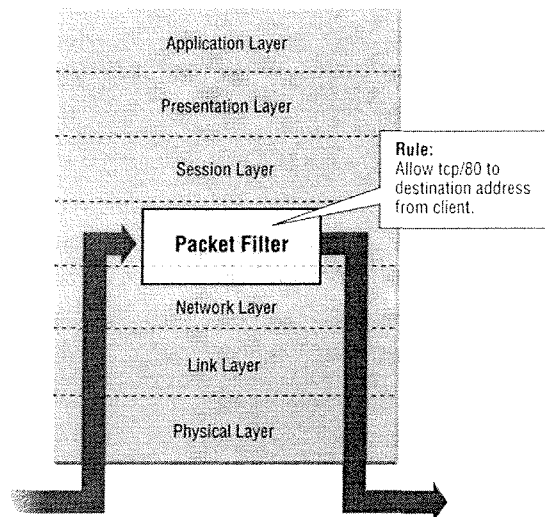


Figure 2.7. A Packet-filtering Device.

In theory, an application-level proxy can make more sophisticated decisions, but it is usually slower than inspecting the IP datagrams. As ‘inspection’ technology gets more sophisticated (it is now becoming possible to keep track of the state of many application-level protocols), the gap between the two approaches lessens. Many firewall products provide both application-level proxies and IP-level inspection or filtering. These products are referred to as hybrids.

(3) Bastion hosts

The bastion hosts are the application servers in the perimeter. A bastion host usually runs one specific piece of software, such as a mail gateway or some web server software. A bastion host has no unnecessary services running, and it is installed and configured in a highly secured manner.

(4) Switches and hubs

As with any other network, you need switches or hubs to build the network infrastructure in the perimeter. A network segment that uses a hub is a shared media, where all traffic is visible from all network stations (hosts). On the other hand, a switch connects the sender directly to the receiver for every Ethernet packet. This provides improved performance for unicast (one-to-one) traffic, but also some additional security. If one of the hosts on a network segment is compromised, an intruder may be able to install a network sniffer to spy on traffic on that segment to get information. However, if a switch is used, the intruder may not be able to see the traffic between other hosts.

I recommend that you use “dumb” switches and hubs without management software if possible. If the switch or hub device has its own IP stack, it may be vulnerable to attacks, and for this reason, it will have to be secured in the same manner as your bastion hosts.

A Perimeter Network Example

The best way to describe how all the components fit together is to present an example perimeter network design. The example network I describe in this section is very general and simplified; don't use it as a ready-to-run implementation blueprint.

What are the objectives of this design?

A design always has to meet some core objectives. These are usually determined by specific business requirements. Let's assume that our example company has the following needs:

- (1) It must allow access to the web servers from the Internet.
- (2) It must accept incoming mail.

- (3) It must allow outbound web and FTP from the internal network.
- (4) It must allow outgoing mail.

The example company must solve these business objectives with regard to two key network security needs:

- (1) No direct traffic can be allowed between the Internet and the internal network.
- (2) If one component in the perimeter is compromised, it should not result in a compromise of the entire perimeter or the internal network.

What's a possible solution to these problems and objectives?

The solution shown in Figure 2.8 meets all the objectives of this example and protects the perimeter both from external and internal threats. The solution implements a perimeter network with the web servers, a firewall gateway, a mail gateway, and an HTTP proxy server.

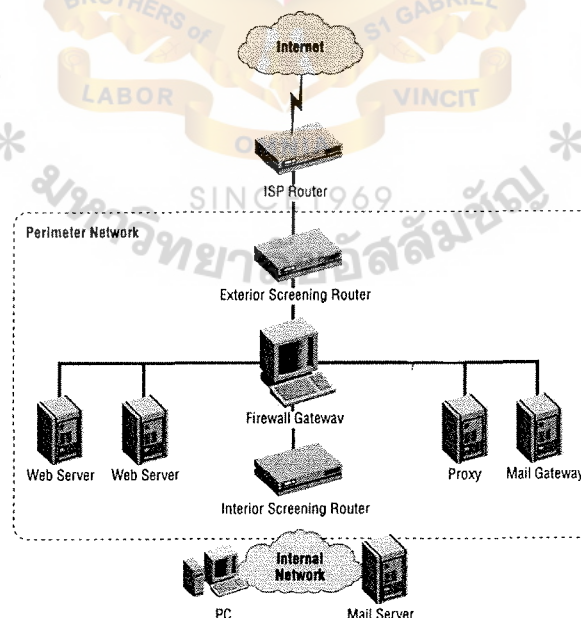


Figure 2.8. Example Perimeter Network.

In this example, the following services are allowed:

- (1) Only inbound HTTP to the web servers and inbound SMTP to the mail gateway are allowed from the Internet.
- (2) The proxy server and mail gateway are allowed to do DNS queries (udp/53).
- (3) The proxy is allowed outbound HTTP, HTTPS, and FTP.
- (4) The mail gateway is allowed to send mail (SMTP) to the Internet and to relay incoming mail to the internal mail server.
- (5) The internal mail server is allowed to relay outgoing mail to the mail gateway.
- (6) The internal network is allowed to use the proxy server in the perimeter.

The screening routers protect the perimeter from the Internet and the internal network from the perimeter in case there is a problem with the firewall gateway. You'll notice that no direct traffic is allowed between the internal network and the Internet and vice versa.

The proxy and mail servers are placed on a different network from the web servers; doing so separates outgoing web surfing from published web services. In the future, the company might consider a separate Internet connection for outgoing web traffic to guarantee bandwidth to its public web servers.

This design has four separate security zones:

- (1) Two zones between the firewall and the screening routers
- (2) One zone for the web servers
- (3) One zone for the mail gateway and proxy server

As a result, the perimeter is well compartmentalized; if one security zone is compromised, the others remain intact. Note that if the firewall gateway is

compromised, multiple security zones are also compromised. However, the interior screening router still protects the internal network.

All the components in the perimeter must be hardened to a very high level. This implies removing all unneeded or insecure services that are provided by default. An easy thing to do is to list the active network services with a command (*netstat -an* on most operating systems), and to scan and probe the host for available services to identify which services you need and which ones you can turn off or remove.

2.3 The Windows NT/2000 Architectures

This section provides a very basic summary of the architecture of Windows NT and Windows 2000 systems. You'll need at least this background information for understanding the instructions in subsequent chapters.

Windows NT is a multithreaded, micro-kernel-based operating system. The term micro-kernel implies that the kernel component is very small, and provides only basic functions such as thread dispatching and hardware exception handling.

Hardware-specific code is kept in a separate layer called the Hardware Abstraction Layer (HAL). The HAL simplifies porting of the operating system to new processor architectures like the IA-64.

The core operating system code runs in privileged processor mode. This mode is also known as protected mode (when referring to the CPU), or kernel mode (when referring to a process or thread). Protected mode provides direct access to system memory and other hardware. Applications run in a nonprivileged processor mode known as user mode and have no direct hardware access. Applications have to use the system calls—the API (Application Programming Interface)—in the underlying operating system to perform tasks such as reading or writing to memory or to the screen.

The basic Windows NT architecture is shown in Figure 2.9.

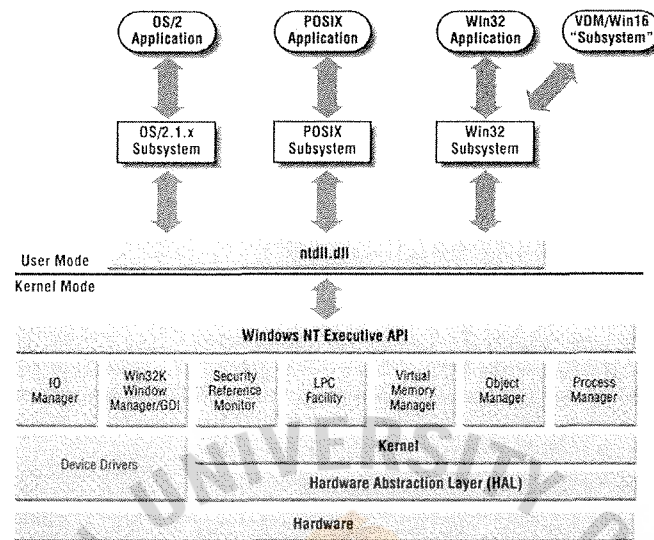


Figure 2.9. The Windows NT Architecture.

Windows NT/2000 Subsystems and Services

Operating system services are kept in discrete subsystems, some running in user mode and others in kernel mode. There are several kernel mode subsystems in Windows NT. They provide NT's native functionality for user mode subsystems through ntdll.dll (running in user mode). The kernel mode subsystems make up the Windows NT Executive, and consist of the following:

(1) Object Manager

The Windows NT architecture is not strictly object-oriented, but internal structures such as shared memory segments, processes, and threads are represented as objects to provide a uniform method for handling things like access control. The Object Manager creates, manages, and deletes

Windows NT Executive objects. Objects are represented in a hierarchical namespace much like a file system.

(2) Process Manager

Responsible for creating and terminating processes and threads using underlying kernel functions.

(3) Virtual Memory Manager

Implements the virtual memory used to allocate a private address space to each process.

(4) I/O Manager

Provides a device-independent I/O system to processes. It dispatches I/O requests to the appropriate device driver.

(5) Local Procedure Call (LPC) Facility

Implements a fast, lightweight version of Remote Procedure Call (RPC) for communication between components within a computer.

(6) Security Reference Monitor (SRM)

Enforces the access and audit policies in the system. The Security Reference Monitor provides access validation, privilege checking, and audit message generation at runtime for both user and kernel mode processes.

(7) Window Manager and Graphical Device Interface (GDI)

These components make up the kernel mode part of the Win32 subsystem. They handle user input and screen output. All of the Win32 subsystem originally ran in user mode; however, for performance reasons, a part of it was moved to kernel mode as of NT 4.0.

The subsystems running in user mode are called the environment subsystems. There are three environment subsystems:

(a) Win32 subsystem

The part of the Win32 subsystem running in user mode. The Win32 sub system is a required part of the operating system and is loaded as a part of the boot sequence. The subsystem consists of the Win32 API DLLS (kernel32.dll, user32.dll, .gdi32.dll) and the Win32 subsystem process (csrss.exe).

(b) POSIX subsystem

Provides support for POSIX.1 applications. It is an optional component that is loaded on demand.

(c) OS/2 1.x subsystem

Provides support for OS/2 1.x console applications. It is an optional component that is loaded on demand.

Windows NT Networking

The first version of Windows NT (Windows NT 3.1) was released in 1993. It was positioned as a successor to the LAN Manager products from Microsoft and IBM. To interoperate and provide backward compatibility with these products, it had to support some established networking standards, such as NetBIOS and SMB. It is important you understand what these protocols are and how they are used in Windows NT. They still provide the foundation for most Windows NT network communication in both Windows NT 4.0 and Windows 2000.

(1) NetBIOS

NetBIOS (Network Basic Input/Output System) is a standard for transmitting data between computers over the network. The NetBIOS specification was developed for IBM back in 1983 to allow network communication between applications. NetBIOS provides three key services:

Name service (port 137)

Locates NetBIOS names on the network.

Session service (port 138,139 or 445)

Provides a connection between two computers.

Datagram service

Provides a connectionless communication channel between computers.

The first implementations of NetBIOS didn't separate the software interface from the network protocol. Later on, the network-level part of the standard was named NetBEUI (NetBIOS Extended User Interface). The Windows NT version of Net BEUI is also referred to as NBF (NetBIOS Frame). Nowadays, NetBIOS can use transports other than the nonroutable NetBEUI protocol, such as TCP/IP (NetBIOS over TCP/IP—NetBT) or IPX/SPX.

(2) **Server Message Block (SMB)**

Remember that NetBIOS is merely a standard for finding resources and transmitting bits. A higher-level protocol is required on top of NetBIOS for it to be of any real use. Here's where SMB comes in. Server Message Block (SMB) is a standard for sending commands and data. SMB is mostly used for file and print sharing, but it can also be used for Inter-Process Communication (IPC) to communicate with processes on other systems.

SMB over NetBIOS uses ports udp/137 (NetBIOS name service) and udp/138 (NetBIOS datagram service) or tcp/139 (NetBIOS session service). Windows 2000 includes support for running SMB without NetBIOS over tcp/445. This support is referred to as Direct Host.

(3) NT networking architecture

The I/O Manager in the NT Executive is responsible for most I/O processing, including disk and network I/O. Figure 2.10 illustrates some of the networking components in the I/O Manager and shows how user mode services interact with these components.

Like all subsystems in the Executive, the I/O Manager exposes a number of APIs to user mode processes. These APIs include the following:

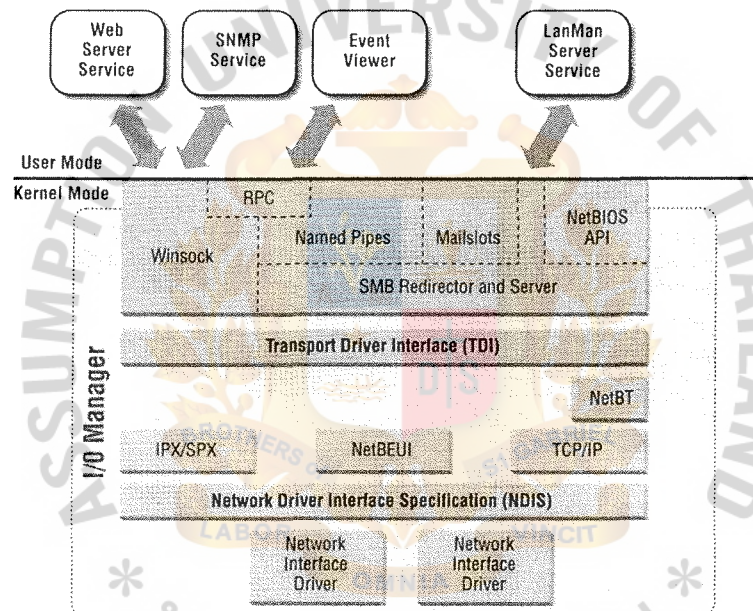


Figure 2.10. The Windows NT Networking Architecture.

Windows Sockets (Winsock)

The Windows NT implementation of the widely used Sockets API. Applications that use Winsock include Internet Explorer, IIS, Telnet, and FTP.

SMB Named Pipes

One-way or duplex communications channels between the pipe server and one or more pipe clients.

SMB Mailslots

A simple IPC mechanism that can be used to send or receive small (less than 425 bytes) datagram broadcast messages.

Remote Procedure Call (RPC)

Microsoft's Remote Procedure Call (MS-RPC) provides a mechanism for using ordinary function calls to communicate with processes on another computer. Distributed Components Object Model (DCOM) components use MS-RPC.

There are two ways of performing RPC communication between two hosts:

MS-RPC over SMB

Uses SMB-named pipes as transport for the RPC calls. Administrative tools such as Server Manager, User Manager, Performance Monitor, and Event Viewer all use MS-RPC over SMB to connect to remote hosts. Windows NT domains also rely on MS-RPC over SMB.

MS-RPC using Windows Sockets

Communication is established by using dynamically assigned high ports (>1023) and the RPC portmapper services tcp/135 and udp/135. This RPC method is often used by DCOM applications.

SMB filesystem drivers

There are two components that enable SMB file sharing:

SMB Redirector

The redirector is a filesystem driver that communicates with the SMB server driver component on a remote system. The Workstation service uses the SMB redirector.

SMB Server

The Server filesystem driver and the Server service work for the connections requested by client-side redirectors, forwarding them to the appropriate local filesystem driver, such as NTFS.

NetBIOS Interface API

The NetBIOS interface API is provided primarily for existing applications that use IBM NetBIOS 3.0 and need to be ported to the Win32 API.

There are two boundary layers in the network architecture:

Transport Driver Interface (TDI)

The TDI provides developers with a protocol-independent network API for network services. Developers need only to program against the TDI to support all available network protocols.

Network Driver Interface Specification (NDIS)

The NDIS wrapper driver communicates with the network protocols. The wrapper driver provides a uniform interface between protocol drivers and NDIS device drivers.

Bindings enable communication between two components on adjacent layers in the protocol stack. For example, bindings can be configured to limit a service to one network protocol or to allow only a network protocol on one of the network adapters in the system.

In the example shown in Figure 2.11, the binding between the Server service and the NetBEUI protocol has been removed. This means the Server is not able to service requests from NetBEUI clients. TCP/IP is bound only to the NIC1 network interface card. IPX/SPX and NetBEUI are bound only to NIC2. As a result, the system will only use TCP/IP on NIC1, and both IPX/SPX and NetBEUI on NIC2.

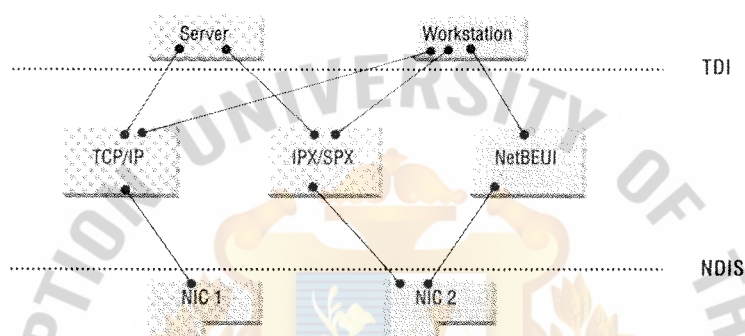


Figure 2.11. Example of Bindings in the Windows Network Architecture.

2.4 Windows NT/2000 in the Perimeter Network

Features like discretionary access control, security auditing, and memory protection place the Windows NT core operating system on par (or better) with many Unix systems in terms of local host security. So why do many people claim that Windows is less secure than Unix?

The problem is not really Windows itself; rather it is the services and applications built on top of the operating system that are the weakest links.

The following sections describe some fundamental principles of secure system design, as well as examine how some of Windows NT/2000's services and applications stack up to these principles.

(1) Least Privilege

A very important principle is that of least privilege. The least privilege philosophy dictates that an application should be designed to run only with the privilege level it needs to execute properly—and no more.

Consider the following question: what privilege level do you need to grant to a web server application? The simplified answer is that the application needs the right to read the data files it serves. Now, take a look at the Internet Information Server's (IIS) WWW service. By design, it has to run as Local System, the highest privilege level in Windows. IIS does run the actual worker threads with lower rights, but if an attacker manages to break IIS before the security context switch is made, he will be able to do anything, including deleting filesystems, starting up a back door, and so on.

Microsoft designed IIS in this way to be able to integrate the web server with the NT security architecture. There is not much specific security code in IIS5; instead, it uses the same access control mechanisms as any other NT process would. The IIS authentication mechanisms use the NT account database. Access to individual files and directories is controlled by NTFS DACLs, just like on a file server. To achieve this level of functionality, IIS needs to be able to start a process or a thread in the security context of the connecting user, and to call the required Win32 APIs it needs to run at very high privilege level.

It is unfortunate that there is no option to install IIS without this functionality, since most Internet web servers don't need user authentication, and because of this, could run IIS at a much lower privilege level.

Unfortunately, many Windows applications and services run as Local System. Some of them may not need that privilege level, but it is the default. Most Windows software vendors don't seem to be aware of the least privilege approach, or at least they don't reflect such awareness in their code. As a result, a bug or back door in these programs can compromise the security of your entire machine. If an application is exposed in the same way it is in a perimeter network, it needs to be designed with security in mind. In fact, the top priority in the perimeter is often not functionality or speed—it is security.

Any application that must run as Local System is potentially a major security hazard. It is a sitting duck waiting for a new buffer overflow attack to happen. If you're running web servers like IIS, one possible solution is to place an application-level proxy in front of those servers. The proxy should be able to verify that any requests to the IIS WWW service conform to the HTTP standards. Any malformed HTTP request will be blocked in the proxy. As a result, the web server is protected from many forms of attack. The disadvantage of having a reverse proxy as an additional layer of security is that it will impact performance to some extent. You also need to make sure that the proxy solution isn't a single point of failure if you want to build a highly available site.

The choice of proxy server product depends on your security needs. It may be sufficient to use an intelligent and configurable application-level (or hybrid) firewall. A better solution may be a combination of one or several dedicated firewalls and a reverse proxy server, as shown in Figure 2.12; such a configuration provides additional layers of security.

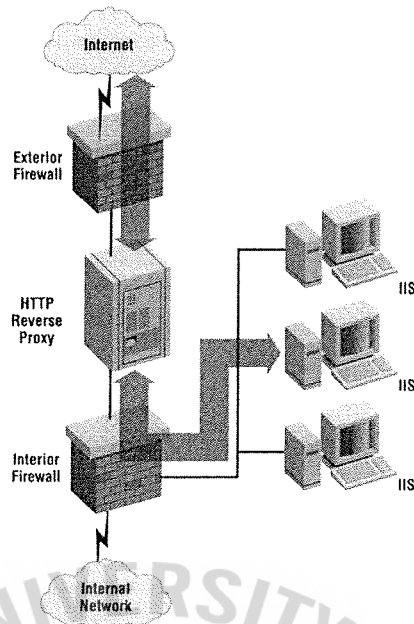


Figure 2.12. Dual Firewall Systems and Reverse Proxy Solution.

(2) Separate Ports

Another important principle of secure design is to use one (or a few) fixed TCP/IP port (TCP or UDP) per application. It is good practice to use different ports for logging, viewing performance data, network logon, and so on. This separation makes it possible to implement granular network access control in the perimeter. It is often a good idea to design an application using this method.

So how does Windows behave on the network? Windows NT is terrible—just about every service uses SMB over the NetBIOS session port (tcp/139). Windows 2000 is somewhat better. Logon authentication is Kerberos (tcp/88 and udp/88). There's also LDAP (tcp/389) to the Active Directory database. Everything else uses SMB just as in Windows NT 4.0—either over NetBIOS or over the new Direct Host protocol port (tcp/445).

Microsoft could have done a better job here. It is virtually impossible to have Windows servers that need to communicate using NetBIOS or Direct Host in different compartments in the perimeter.

If you plan to support a large number of Windows NT 4.0 or Windows 2000 servers, you may find that it is difficult to manage them as separate hosts. It is tricky to have both security and a management platform that scales up to hundreds of servers. It is tempting to use a centralized accounts database (using NT domains) and other NetBIOS-based tools like Event Viewer and Server Manager.

At very large sites (those with 50 or more NT servers in the perimeter), a common setup is to have dual-homed NT/2000 systems with NetBIOS/SMB unbound (deactivated) from the external network interface. The internal network interface is connected to a kind of management network so that the servers can be managed using the standard RPC-based tools in a domain environment. A remote console solution such as Terminal Server is often used to gain remote access to the management network. Figure 2.13 shows such a solution.

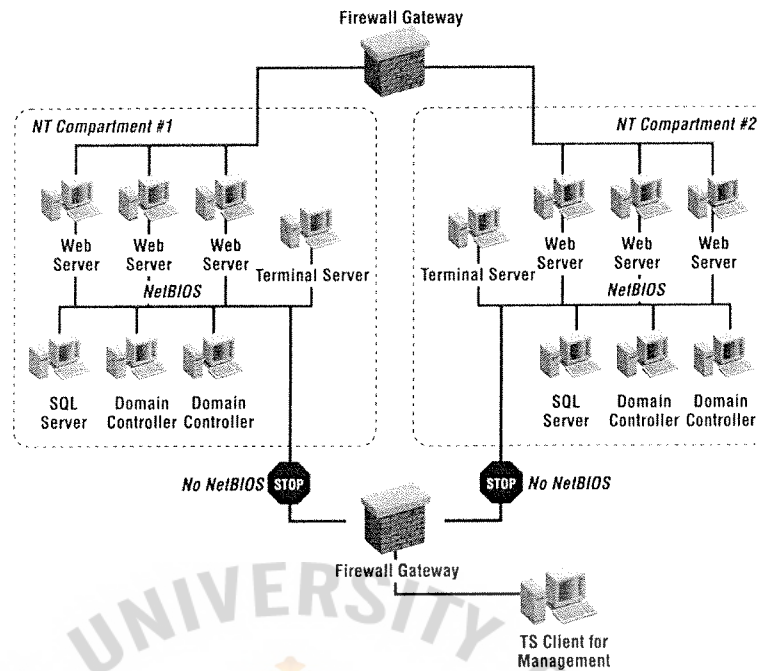


Figure 2.13. NT Domains in the Perimeter.

If you must run NetBIOS in this manner, you should be aware that it will be extremely difficult to build a well-compartmentalized perimeter. You can't set up network access control to allow only a certain type of NetBIOS traffic. As a result, you have to consider all hosts within an NT domain as one security zone. In such a configuration, if one server is broken into, there are no security mechanisms that can protect your domain controllers and other servers in the same administrative domain.

NetBIOS was not designed with security in mind and I recommend against using it in a perimeter. However, if you choose to implement NetBIOS anyway, I urge you to implement an extremely secure perimeter using multiple firewalls and a reverse proxy solution. Do not allow any direct connections from the Internet to your exposed services.

2.5 Cryptography Basics

This section is a very brief summary of some of the terms and algorithms relevant to discussions in later chapters. For more detailed information on this complex topic, consult a good cryptography reference.

Public Key Cryptography

With public key cryptography, each party has a key-pair consisting of a private key and a public key. The public key is published while the private key is kept secret. Data encrypted with the public keys can only be decrypted using the private key and vice versa. Public key cryptography can also be used for authentication (through the use of digital signatures).

An important advantage of public key cryptography is that there are less complicated key distribution problems. All parties that want to be able to communicate using public key cryptography need to publish their public key in some kind of directory. When Alice wants to send an encrypted message to Bob, she uses Bob's public key to encrypt the data. Alice can also digitally sign a message by encrypting it with her private key. Bob can then decrypt the message using his private key and verify Alice's signature by decrypting using her public key.

The main disadvantage is that public key cryptography is slow compared to symmetric key cryptography.

Two common public key systems are:

- (1) Rivest-Shamir-Adleman (RSA)

The RSA cryptosystem is the most widely used public key cryptosystem. It is designed for doing both digital signatures and data encryption. RSA is used in Internet standards and drafts like IPSec,

S/MIME, and TLS (the successor to SSL). RSA was patented in the U.S. until September 20, 2000.

(2) Digital Signature Algorithm (DSA)

The Digital Signature Algorithm is an unpatented alternative to RSA. DSA was intended for performing digital signatures only, but it can be adapted for encryption as well. DSA is described in the Digital Signature Standard (DSS). DSA was designed by the National Security Agency (NSA) based on the ElGamal algorithm, which is unpatented.

Symmetric Key Cryptography

With symmetric key cryptography, a single key is used for both encryption and decryption. Symmetric key cryptography is also known as secret key cryptography. The two most common methods that use symmetric key cryptography are stream ciphers and block ciphers, described in the following sections.

An important advantage of symmetric cryptography is that it is generally faster than public key cryptography. The main disadvantage is that it is hard to manage and distribute the keys to all parties in a secure manner.

(1) Stream ciphers

Stream ciphers use a key stream that is the same length as the cleartext (unencrypted) data to produce the ciphertext (encrypted) data. The key stream can be independent of the data, or it can be generated based on it. Stream ciphers can be designed to be extremely fast. The most commonly used stream cipher is probably RC4 (Rivest Cipher 4), which is used in SSL-enabled browsers, for example.

(2) Block ciphers

Block ciphers use a fixed-size encryption key to encrypt fixed blocks (generally 64 bits) of data. The following are some commonly used block ciphers:

(a) Data Encryption Standard (DES)

The U.S. Government's Data Encryption Standard is a block cipher, originally created by IBM, that operates on 64-bit blocks using a 56-bit key. This version is known as Single DES. The latest revision of DES now incorporates the Triple DES (3-DES) algorithm. This is also referred to as the TDEA (Triple Data Encryption Algorithm). 3-DES is simply Single DES performed three times (encrypt, decrypt, encrypt) with three different keys (3 x 56-bit keys) on the same 64-bit block—hence the name 3-DES.

(b) Advanced Encryption Standard (AES)

The Advanced Encryption Standard, currently under development, will eventually replace DES as the U.S. government encryption standard. The AES development effort is led by the U.S. National Institute for Standards and Technology (NIST). NIST initiated the AES effort in 1997 as a “call for algorithms” in which the cryptographic community was invited to submit AES candidates. On October 2, 2000, the Rijndael (pronounced Rhine-doll) algorithm was declared the winner. It will eventually become the official AES in 2001.

(c) Blowfish

Bruce Schneier's Blowfish algorithm is another block cipher that operates on 64-bit blocks. Blowfish uses variable-length keys (32 to 448 bits) and offers very good performance. Blowfish is unpatented and free. You can get it from <http://www.schneier.com/code/Blowfish.c>

(d) International Data Encryption Algorithm (IDEA)

The International Data Encryption Algorithm is yet another block cipher that operates on 64-bit blocks. Many consider IDEA the best block cipher algorithm to date. It is used in several products and protocols such as PGP and some SSH implementations. Ascom Systec Ltd. (<http://www.ascom.ch/infosec/idea.html>) holds the rights to the IDEA algorithm. The licensing cost is about \$10 per end user.

Hash Algorithms

Hash algorithms provide a method for reducing variable-length data to a small fixed-length hash. Hashes are also called message digests or fingerprints. Hash algorithms are often used to produce message integrity checksums or to store passwords in a secure manner. A hash algorithm has the following requirements:

- (1) It should not be possible to deduce the data from the hash.
- (2) No sets of data should produce the same hash.
- (3) It should not be possible to generate a given hash.

Two of the more common hash algorithms are:

(1) Message Digest (MD5)

MD5 is available from RSA Data Security. Ronald Rivest (the "R" in RSA) published the MD5 hash algorithm in 1992 as RFC 1321. MD5 was an improvement over the previously published MD4 algorithm, which has

some design weaknesses. MD5 produces a 128-bit hash from an input of any length.

(2) Secure Hash Algorithm (SHA-1)

Available from the U.S. National Institute of Standards and Technology, SHA-1 is the NIST Hash Standard. It is considered to be the most secure hash secure algorithm today. It produces a 160-bit hash from a variable-length input.



III. WINDOWS 2000 VS WINDOWS NT 4.0

Microsoft has taken strides with Windows 2000 to improve the ease of securing the OS. In addition, it has implemented features that are compatible with cutting-edge security standards that make interoperability simple and extension of industrial-strength security solutions fairly easy. This chapter is dedicated to a discussion of the following built-in features and tools:

- (1) Security Templates and Security Configuration Analysis
- (2) Group Policy
- (3) IPSec
- (4) Kerberos
- (5) Encrypting File System (EFS)
- (6) Runas
- (7) A Windows File Protection (WFP)

This list is by no means a comprehensive catalog of all of the security-related functionality of Windows 2000; rather, it shows what I view as the key new (or significantly updated) features of the OS. In addition, while I'm not going to cover each of these entities exhaustively, I will focus specifically on how they can be used to counter the attacks. Truly, these are the tools that will allow you to raise the bar for attackers and ease the burden for security administrators when running Windows 2000.

3.1 Security Templates and Security Configuration and Analysis

Introduced in NT4 Service Pack 4 as an optionally installed component, Security Templates and Security Configuration and Analysis are probably among the best timesaving tools you can use to deploy security across your Windows 2000 infrastructure, especially when leveraged in conjunction with Group Policy.

Security Templates are structured lists of security-relevant Windows 2000 settings that can be edited and applied to a system at the click of a mouse, bypassing the need to identify, locate, and configure the dozens of individual security settings. In addition, these template files can be compared to the current settings of a given system, showing configurations that are in compliance or not (the Analysis part of the equation).

Security Templates and Security Configuration and Analysis can be accessed most easily by bringing up a blank Microsoft Management Console (MMC) window and adding the Security Templates and Security Configuration and Analysis snap-ins, as shown in Figure 3.1 Let's examine Security Templates and then Security configuration and Analysis to illustrate the power of these tools.

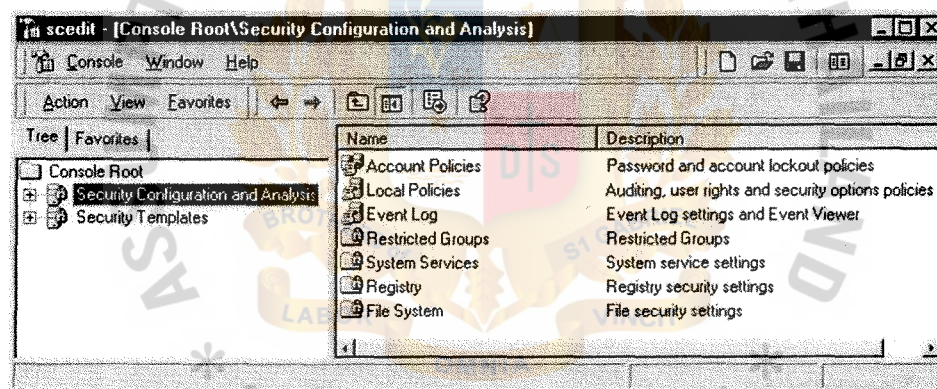


Figure 3.1. An MMC Window with the Security Configuration and Analysis and Security Template Snap-ins Installed.

(1) Security Templates

The Security Templates node in the left pane of Figure 3.1 is set by default to browse the `%systemroot%\security\templates` directory, where the default Windows 2000 Security Templates are kept. You can click one of

the Security Templates to examine it more closely, which will illustrate the aspects of Windows 2000 that can be configured:

- (a) Account Policies Equivalent to the Windows 2000 Security Policy settings of the same name; includes password, account lockout, and Kerberos policies
- (b) Local Policies Equivalent to the Windows 2000 Security Policy settings of the same name; includes auditing, user rights assignment, and security options (where most of the critical settings lie) policies
- (c) Event Log Configures Event Log settings
- (d) Restricted Groups Defines the only authorized members of groups, such that any unspecified members are removed when the policy is applied (a good way to ensure that attackers don't plant backdoor accounts in Domain Administrator some other powerful group if applied via Group Policy)
- (e) System Services Defines the startup behavior of services and access control permissions (allowing you to disable specific services, for example)
- (f) Registry Defines Registry key access control settings
- (g) File System Defines file-system access control settings

Although they do not cover every aspect of the Windows 2000 operating system, and the ability to define settings not already in the template (such as adding a Registry value) is limited, Security Templates clearly offer a great shortcut for administrators faced with manually configuring many different installations of Windows 2000 securely and consistently.

Table 3.1. Definition of the Default Windows 2000 Security Templates.

Template	Definition
setupsecurity	Default, out-of-the-box security settings; apply this template to back off more stringent security settings applied by other template for whatever reason
Compatws	Relaxed security from default clean Windows 2000 Professional install
Basicdc	Default security Settings for a domain controller
Basicsv	Default security Settings for a server
Basicwk	Default security Settings for Windows 2000 Professional
Securews	Improve security of additional areas over basicwk
Securedc	Improve security of additional areas over basicdc
Ocfilesw	Applies more secure configuration to optionally-installed Windows 2000 Professional components(apply in addition to securews or hisecws)
Ocfiless	Applies more secure configuration to optionally-installed Windows 2000 Server components(apply in addition to securedc or hisecdc)
Hisecdc	More secure Windows 2000-only enhancement beyond securewe
Hisecws	More secure Windows 2000-only enhancement beyond securews

As visible within the Security Templates MMC node are 11 pre-defined templates shipped with Windows 2000. Table 3.1 provides a quick guide to what each of the default Windows 2000 templates defines, roughly

in order of lowest security to highest (ocifies templates are exceptions to this scale).

Note that with the exception of ocfilesw and ocfiless, these templates are meant to be applied cumulatively—i.e., basicwk or basicsv should be applied first, securews or securedc should be applied next, and finally hisecws or hisecdc should be installed. The reason for this is that each of these templates configures specific areas of the OS (file and Registry ACLs, user rights, group memberships, policies, audit settings, and so on). For example, basicwk brings all areas up to a basic level of security, while securews provides increased security only for areas of the operating system that are not covered by permissions, including increased security settings for the account policy, increased settings for auditing, and increased security settings for some well-known security-relevant Registry keys. Access control lists (ACLs) are not modified by securews, because the assumption is that default Windows 2000 security settings are in effect. Similarly, you should use the ocfile templates if you have installed optional components on either Windows 2000 Professional or Server.

As Table 3.1 illustrates, the most secure of the default templates is hisecws. Microsoft also provides a template called hisecweb via its Web site. Recognize, however, that these are just templates—examine the configurations in these files carefully for compatibility with your applications, note which settings could be made more stringent, and be careful to observe that many additional settings could be added to tailor security to your needs. It is easy to build your own template by simply right-clicking the default template of your choice (hisecws, for example) and

selecting Save As. Then you can go back and configure each setting the way you want within the new template.

One good example of a setting that should be included in the hisecws template is replacing the NTFS ACLs on the cmd.exe shell and other powerful administrative tools in `%systemroot%\system32`. This can be accomplished easily using Security templates.

Next, I will talk about the Security Configuration and Analysis node, where I will apply the settings defined in a template or audit a system against a template.

(2) Security Configuration and Analysis

Single-clicking on the Security Configuration and Analysis node in the left pane of the MMC window shown in Figure 3.1 will cause the right pane to prompt the user with instructions for how to create a database before proceeding with configuration and/or analysis. The database is a temporary storage place for holding security template information and analysis results. Follow the prompts to open a new database, and import one of the built-in security templates (one of the default templates or one of your own design).

After a template has been imported into the database, you can use it to analyze or configure the local system by right-clicking the Security Configuration and Analysis node in the left pane of the MMC window and selecting either Analyze Computer Now or Configure Computer Now.

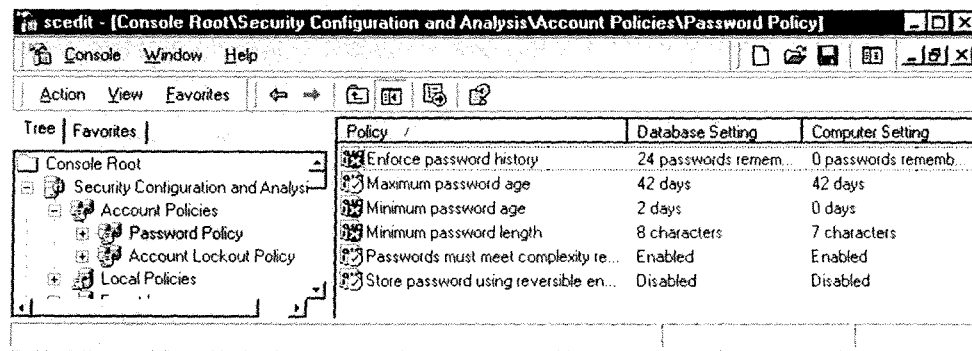


Figure 3.2. The Results of a Security Analysis Showing Current Computer Settings That Match the Selected Template (Green Checkmark Icons) or Mismatch (Red X Icons).

If you select the Analyze option, a dialog will prompt you for a location to save a log of the analysis process (the default path is *%userprofile%\Local Settings\Temp\ template.log*), and then a progress bar will appear as Windows compares the current settings on the local computer to those imported into the database from the template. When this process is complete, you can select any item to determine whether it matches the template. As shown in Figure 3.2, out-of-compliance settings are shown with a red X icon and matching settings are indicated by a green checkmark icon. If no preference was specified by the database/template, no icon appears and the Database Setting is indicated as Not Defined (the log calls this “Not configured”). The log of the analysis session also contains a record of each comparison result, including the date and time when the analysis was run, each check performed, the result of each check (whether the setting was analyzed or mismatched, or an error resulted in querying the value), and so on. The log does not include records of settings that match the template, unfortunately. To access the log, you can right-click the Security Configuration and Analysis node and select View Log.

The Configure option works in much the same way that Analyze works, but instead of simply comparing current machine settings to the database/template, the settings are actually applied.

You may have noticed so far that I've discussed Security Configuration and Analysis only as it applies to a single machine. Obviously, analyzing and configuring Windows 2000 would be much more efficient if it could be done simultaneously over the network across many systems. There are two ways to do this: the old-fashioned way and via Group Policy.

The old-fashioned way is to use the *secedit* command-line tool to perform the analysis or configuration via a logon script or some other distributable batch mechanism. (Run the *secedit* command with no arguments and the help system will pop up.) If a Windows 2000 domain is available, though, a more powerful option is available: Group Policy.

3.2 Group Policy

One of the most powerful new tools available under Windows 2000, Group Policy can be used to affect much more than just security settings. In this chapter, I will focus solely on its security-related functionality.

Group Policy is Windows 2000's centralized configuration management architecture. It is implemented by Group Policy Objects (GPOs), which define configuration parameters that can be applied (or linked) to users or computers. There are two types of GPOs: the Local GPO (LGPO) and Active Directory GPOs (ADGPOs).

The LGPO is stored in and comprises several files: *gpt.ini*, administrative templates (.*adm*), security configuration files (.*pol*), and logon/logoff and startup/shutdown scripts. ADGPOs are stored in *%systemroot%\system32\sysvol*

<domain>\Policies, and a pointer to each ADCPO is also stored in the directory in the System Policy container. As you might guess, the LGPO applies only to the local computer. ADGPOs can be applied to sites, domains, or organizational units (OUs), and multiple GPOs can be linked to a single site, domain, or OU.

Of particular interest to us are the security-relevant settings of a GPO, grouped under the Computer Configuration\Windows Settings\Security Settings node. The settings available here mirror those available via the Local Security Settings applet (in fact, Local Security Settings is simply a shortcut interface to the Local Computer GPO's Computer Configuration\Windows Settings\Security Settings node). As we have seen, the Security Settings node defines account policies; audit policies; and event log, public key, and IPsec policies. By allowing these parameters to be set at the site, domain, or OU level, the task of managing security in large environments is greatly reduced. Even better, Security Templates can be imported into a GPO. Thus, Group Policy is the ultimate way to configure large Windows 2000 domains securely.

3.3 IPsec

One of the bolder inclusions in Windows 2000 is support for the evolving IPsec standard, IPsec (RFCs 2401, 2402, and 2406). IPsec specifies a mechanism for achieving end-to-end security of IP datagrams, including authentication, confidentiality, integrity, and anti-replay services, without requiring intermediate devices to understand the protocol.

Because of IPsec's history as a communications security protocol, many people associate IPsec with encrypted network packets—Virtual Private Networks (VPNs) and so on. The Windows 2000 IPsec implementation also provides a fairly simple mechanism for filtering unicast IP packets, much like a host-based firewall.

Advantages of IPSec Filters

Why use IPSec filters? There are many good reasons. First, the IPSec filtering functionality is built into the OS, so it is available wherever Windows 2000 is deployed. Second, filters are fairly easily crafted by a knowledgeable network security administrator and can be applied with a simple mouse-click or via a batch script (no reboot required). Third, IPSec filters beat the older TCP/IP security feature hands down because they don't require a reboot and they actually block ICMP traffic (a little-known factoid about TCP/IP security is that ICMP is never actually blocked even if specified in the interface).

IPSec filters can be used in addition to network firewall devices and disabling unnecessary services to supply valuable "defense-in-depth" at the host level in Windows 2000.

Known Limitations of IPSec Filtering

Windows 2000's IPSec implementation was designed as an administrative security tool to provide permit, block, and automatic negotiation of cryptographic protection actions for unicast IP traffic, and to make this basic capability easy to manage on a large scale. It was not designed to be a general-purpose packet filter tool or a comprehensive filtering firewall capability. Microsoft understands that its customer base has seized on this functionality, however, and is working toward providing more comprehensive filtering in future Windows. However, in the current version, IPSec filtering has some limitations that arise from these focused design goals.

A key concept to understand about IPSec is that, by design, certain types of protocols cannot be secured by IPSec. For one, non-IP protocols like IPX and NetBEUI obviously cannot be secured by IPSec. Also, lower layer protocols like Address Resolution Protocol (ARP) are also outside of the bounds of IPSec protection (and

before you go thinking that's not a big deal, realize that SMB can be implemented over Layer 2).

In addition, the so-called default exemptions cannot be secured by Windows 2000's IPSec design. KB Article Q253169 discusses traffic types that by default bypass IPSec filters (the following material is quoted from the Knowledge Base article):

Broadcast

Traffic going from one sender to many receivers that are unknown to the sender. This type of packet cannot be classified by IPSec filters. For example, a standard class C subnet using 192.168.0.x would have a broadcast address of 192.168.0.255. Your broadcast address depends on your subnet mask.

Multicast

As with Broadcast traffic, one sender sends an IP packet to many receivers that are unknown to the sender. These are addresses in the range from 224.0.0.0 through 239.255.255.255.

Resource Reservation Protocol (RSVP)

This traffic uses IP protocol 46 and is used to provide Quality Of Service (QoS) in Windows 2000. Exemption of RSVP traffic is a requirement to allow QoS markings for traffic that may be secured by IPSec.

Internet Key Exchange (IKE)

IKE is a protocol used by IPSec to securely negotiate security parameters (if the filter action indicates that security needs to be negotiated) and establish shared encryption keys after a packet is matched to a filter. Windows 2000 always uses a LDAP source and destination port 500 for IKE traffic.

Kerberos

Kerberos is the core Windows 2000 security protocol typically used by IKE for IPSec authentication. This traffic uses a UTP/TCP protocol source and destination port 88. Kerberos is itself a security protocol that does not need to be secured by IPSec. The Kerberos exemption is basically this: If a packet is TCP or UDP and has a source or destination port = 88, permit.

If you read these exemptions carefully, you may note a few immediately obvious lines of attack against a system protected by even the most stringent IPSec filters.

3.4 Kerberos

As is probably well-known by now, Windows 2000 includes support for an alternative to the traditional NTLM authentication architecture used by previous versions of Windows (a traditional LM and NTLM authentication). Kerberos v5 is a well-established standard that is considered quite secure in professional and academic security circles.

The primary thing to realize about Kerberos is that it is currently much more secure than LM and NTLM protocols because it is not vulnerable to the LM eavesdropping attacks. Unfortunately, there is no way to force Kerberos authentication to be used, as exists for forcing higher security NTLM protocols using the LM Authentication Level setting in Security Policy. Currently, Kerberos will be used if it is available, but otherwise, Windows 2000 silently reverts to NTLM and is again potentially vulnerable to SMB sniffing attacks.

Also recognize that Windows 2000 Kerberos can be used only if a Windows 2000 domain is available. A Windows 2000 domain controller is the only entity that runs the Kerberos services, so it is impossible for a stand-alone Windows 2000 system to authenticate using Kerberos unless it first obtains a ticket from a domain controller.

Some other requirements for Windows 2000 Kerberos are these:

- (1) Both machines (client and server) must reside in the same forest.
- (2) The client must address the server by its DNS or machine name. Kerberos will not be used if one machine contacts the other via its IP address (for example *net use \\IPaddr\c\$*).

One workaround to require Kerberos to be used within a Windows 2000 domain is to assign the IPsec policy Secure Server (Require Security) to all servers within the domain via Group Policy. This will force clients to first authenticate to the domain using Kerberos before communicating via IP with these servers. IIS5 intranet servers can also use Kerberos via the new Negotiate HTTP header with IE5.x clients.

If you do implement Kerberos in your environment, a handy tool to have is the kerbtray utility from the Windows 2000 Resource Kit. The kerbtray utility will catalog any Kerberos tickets resident on the local machine, as shown in Figure 3.3.



Figure 3.3. The Kerbtray Utility Show Kerberos Tickets Held by the Local Machine.

3.5 Encrypting File System

One of the major security-related centerpieces of Windows 2000 is the Encrypting File System (EFS). EFS is a public-key cryptography—based system for transparently encrypting on-disk data in real time so that attackers cannot access it without the proper key. In brief, EFS can encrypt a file or folder with a fast, symmetric encryption algorithm using a randomly generated file encryption key (FEK) specific to that file or folder. EFS uses the Extended Data Encryption Standard (DESX) as the encryption algorithm. The randomly generated FEK is then itself encrypted with one or more public keys, including those of the user (each user under Windows 2000 receives a public/private key pair) and a key recovery agent. These encrypted values are stored as attributes of the file.

Key recovery is implemented in case users who have encrypted some sensitive data leave an organization or their encryption keys are lost, for example. To prevent unrecoverable loss of the encrypted data, Windows 2000 mandates the existence of a data recovery agent for EFS—EFS will not work without a recovery agent. Because the FEK is completely independent of a user's public/private key pair, a recovery agent may decrypt the file's contents without compromising the user's private key. The default data recovery agent for a system is the local Administrator account.

Although EFS can be useful in many situations, it probably doesn't apply to multiple users of the same workstation who may want to protect files from one another. That's what NTFS file system access control lists (ACLs) are for. Rather, Microsoft positions EFS as a layer of protection against attacks where NTFS is circumvented, such as by booting to alternative OSes and using third-party tools to access a hard drive, or for files stored on remote servers. In fact, Microsoft's white paper on EFS specifically claims that "EFS particularly addresses security concerns raised by tools

available on other operating systems that allow users to physically access files from an NTFS volume without an access check.”

3.6 Runas

To UNIX enthusiasts, it may seem like a small step for Windowskind, but at long last Windows 2000 comes with a native switch user (*su*) command called *runas*. As has long been established in the security world, performing tasks under the context of the least-privileged-user account is highly desirable. Malicious Trojans, executables, mail messages, or remote Web sites visited within a browser can all launch commands with the privilege of the currently logged-on user, and the more privilege this user has, the worse the potential damage.

Many of these malicious attacks can occur during everyday activities and are thus particularly important to those who require Administrator privileges to perform some portion of their daily work (adding workstations to the domain, managing users, hardware—the usual stuff). The unfortunate curse of poor souls who log on to their systems as Administrator is that they never seem to have enough free time to log on as a normal user, as security best practices dictate. This can be especially dangerous in today’s ubiquitous Web-connected world. If an administrator comes across a malicious Web site or reads an HTML-formatted email with embedded active content, the damage that can be done is of a far greater scale than if it Joe User on his stand-alone workstation had made the same mistake.

The *runas* command allows everyone to log in as a lesser-privileged user and then to escalate to Administrator on a per-task basis. For example, say Joe is logged in as a normal user to the domain controller via Terminal Server, and he needs to change one of the Domain Admins passwords (maybe because one of them just quit and stormed out of the operations center in a huff). Unfortunately, he can’t even start Active Directory

Users and Computers as a normal user, let alone change a Domain Admin password.

Runas to the rescue! Here's what he'd do:

- (1) Choose Start I Run, and then enter *runas /user:mydomain\Administrator*
"mmc %windir%\system32\dsa.msc"
- (2) Enter the Administrator's password.
- (3) Once Active Directory Users and Computers starts (dsa.mmc), he can change the Administrator password at his leisure, under the privileges of the mydomain\Administrator account.
- (4) He then quits AD Users and Computers and goes back to life a simple user.

Joe has just saved himself the pain of logging out of Terminal Server, logging back in as Administrator, logging back out, and then logging back in as a normal user. Least privilege—and efficiency—rule the day.

One of the more obvious examples of smart use of runas would be to run a Web browser or email reader as a less-privileged user. This is where runas gets tricky, however, as a rather lengthy thread on the NTBugtraq mailing list detailed at the end of March 2000 (<http://www.ntbugtraq.com>). In this thread, it was debated exactly what privileges would trump when a URL was called within a browser window on a system with multiple open windows, including some with *runas /u:Administrator* privilege. One suggestion was to put a shortcut to the browser (minimized) in the Startup group, so that it always started with least privilege. The final word on using runas in this way, however, was that with applications started via dynamic data exchange (DDE), such as Internet Explorer, key security information is inherited from the creating (parent) process. Thus, runas is never actually creating the IP processes needed to handle hyperlinks, embedded Word documents, and so on. Parent process creation varies by program, so actual ownership is difficult to determine. Maybe Microsoft will someday

clarify whether this is actually a more secure practice than completely logging off of all Administrator windows to do any browsing.

Runas is not a silver bullet. As pointed out in the NTBugtraq thread, it “mitigates some threats, but exposes some others” (Jeff Schmidt). It also does not permit per-application restrictions on usage like the UNIX *sudu* command—if runas is available, it can be used against any executable available to the local machine. Use it wisely.

3.7 Windows File Protection

Windows File Protection (WFP) verifies the source and version of a system file before it is initially installed (WFP is sometimes referred to as System File Protection, SFP). This verification prevents the replacement of protected system files with extensions such as .sys, .dll, .ocx, .ttf, .fon, and .exe. WFP runs in the background and detects attempts by other programs to replace or move a protected system file. WFP also checks a file’s digital signature to determine whether the new file is the correct version.

If the file is not the correct version, WFP replaces the file from a backup stored by default in the `%systemroot%\system32\dlcache` folder (this directory is hidden as a protected operating system file), network-install location, or from the Windows 2000 CD. If WFP cannot locate the appropriate file, it prompts the user for the location. WFP also writes an event noting the file replacement attempt to the Event Log.

By default, WFP is always enabled and allows only protected system files to be replaced when installed by the following processes:

- (1) Windows 2000 Service Packs using update.exe
- (2) Hotfix distributions using hotfix.exe
- (3) Operating system upgrades using Winnt32.exe
- (4) Windows update
- (5) Windows 2000 Device Manager/Class Installer

To check the integrity of the WFP-protected files, use the graphical File Signature Checker (sigverif.exe) or the command-line System File Checker (sfc.exe). Notice that several files have no signature—although these could simply be files installed by non-Microsoft software products, some of them appear to have suspicious names.

WFP identifies which files are valid via a mechanism called driver signing. Almost all Windows 2000 files are signed by Microsoft, and the signatures (SHA-1 hashes) are kept in the catalog files (not in the drivers themselves) in the %systemroot%\system32\CatRoot directory. Opening the .cat files here will display the signature information.

3.8 Summary

Compared to Windows NT 4.0, Windows 2000 server was mainly improved on the ease of securing the Operating System. Followings are the seven added-tool functions.

- (1) Security Templates and Security Configuration Analysis
 - (a) Security Templates are the security-relevant lists of Windows 2000 settings. These settings can be edited and applied to a system to locate and configure the individual security settings, and bypass the identified needs.
 - (b) Security Configuration Analysis helps comparing and analyzing the secured-policy settings with others.

- (2) Group Policy

Group Policy is the centralize configuration management architecture of Windows 2000. It is implemented by Group Policy Objects (GPOs), which define configuration parameters that can be applied (or linked) to users or computers.

(3) IPsec

IPsec specifies the mechanism for achieving end-to-end security of IP datagrams, including authentication, confidentiality, integrity, and anti-replay services, without requiring intermediate devices to understand the protocol. Moreover, IPsec filters can be used in addition to network firewall devices and disabling unnecessary services to supply valuable “defense-in-depth” at the host level in Windows 2000.

(4) Kerberos

Kerberos is known as an alternative to the traditional NTLM authentication architecture used by previous versions of Windows, the traditional LM and NTLM authentication. Kerberos v5 is the well-established standard, considered as a quite secure in professional and academic security circles.

(5) Encrypting File System (EFS)

EFS is a public-key cryptography—based system for transparently encrypting on-disk data in real time so that attackers cannot access it without the proper key and particularly addresses security concerns raised by tools available on other operating systems that allow users to physically access files from an NTFS volume without access check.

(6) Runas

The *runas* command allows everyone to log in as a lesser-privileged user to escalate to Administrator on a per-task basis.

(7) Windows File Protection (WFP)

Windows File Protection (WFP) is a tool to verify the source and version of a system file before initial installing. This verification prevents

the replacement of protected system files with extensions such as .sys, .dll, .ocx, .ttf, .fon, and .exe.



IV. SECURING WINDOWS 2000 SERVERS

4.1 Installation and Configuration

This part explains how to build a secure Windows 2000 bastion host by modifying the default Windows 2000 configuration. If you carefully apply the changes suggested in this part, you should be able to build a very secure Windows server. However, there are several caveats.

First, remember that a chain is only as strong as its weakest link. If you follow the configuration steps in this part, your operating system will be quite secure. However, every application installed will also have to be configured with extreme caution—especially applications that require high privileges to run properly.

Second, note that if you secure your Windows 2000 server as described in this part, you'll no longer be able to perform remote administration via NetBIOS-based tools such as Server Manager and Event Viewer. You won't even be able to copy files to the system over the network anymore. However, there's no need to panic! In the next part, Setting Up Secure Remote Administration, I show you how to build a new and more secure remote management foundation for your bastion host.

Third, the methodology presented in this part primarily targets small to medium-sized perimeters with up to 50 or so Windows 2000 hosts. If you have plans to build a perimeter with more than 50 to 75 hosts, consider the dual-homed approach shown in Figure 2.13, and skip the RPC/NetBIOS/SMB steps presented in this part.

The following steps are required to build a bastion host:

- (1) Install a minimal operating system and the latest service pack.
- (2) Remove or disable unneeded components from the operating system.
- (3) Harden the remaining operating system components.
- (4) Set very restrictive permissions on files and other objects.

The rest of this part explains in detail how to perform these actions.

Installation

The installation process on Windows 2000 is started in the usual way. Follow these steps:

- (1) Accept the defaults until you get to the Network Settings dialog box.
- (2) In the Network Settings dialog box, click “Custom” and deselect everything but “Client for Microsoft Networks” and TCP/IP. Uncheck (unbind) “Client for Microsoft Networks” from all interfaces, as shown in Figure 4.1. The reason that you must keep the client software is that the NT/LanMan Security Support Provider (NTLM SSP) is a part of this client software. If you want to run the Internet Information Server (IIS), you must have the NTLM SSP installed; otherwise, I will not start. If you do not plan to run IIS, you can choose to uninstall the “Client for Microsoft Networks.”
- (3) Disable NetBIOS over TCP/IP in the TCP/IP Properties dialog box.

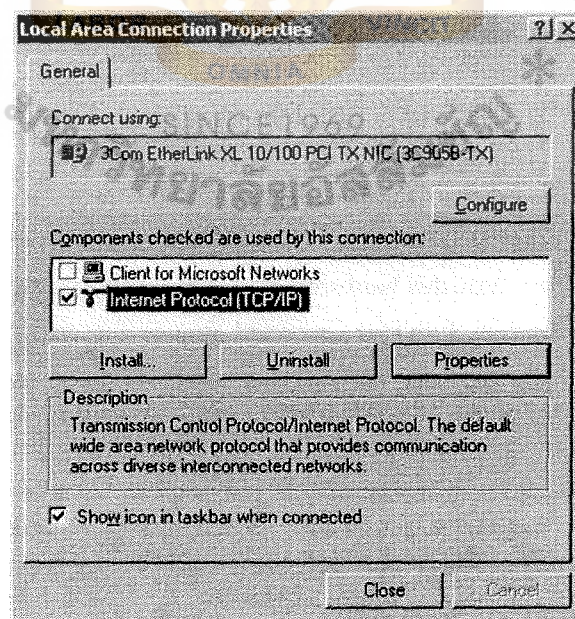


Figure 4.1. The Windows 2000 Configuration and Bindings for a Network Card.

Installing Service Packs and Hotfixes

Microsoft has abandoned the FTP download method for service packs and hotfixes for Windows 2000. Instead, they distribute product updates through their Windows 2000 web site (<http://www.microsoft.com/windows2000/downloads/default.asp>).

Basic Configuration

After installing the operating system, the latest service pack, and all relevant hotfixes, the next step is to ensure that the bastion host has rudimentary security in place. This involves some GUI-based configuration steps that remove unneeded or possibly unsafe components from the operating system.

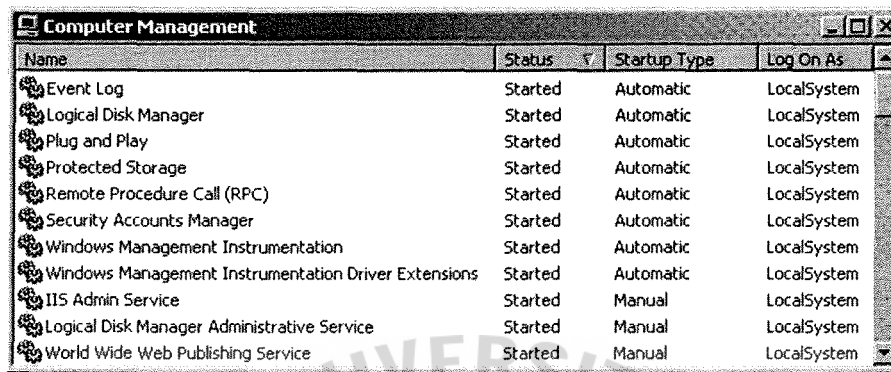
Configuring Windows 2000 Services

Windows 2000 contains almost twice as many services as Windows NT. Fortunately, they are all described in the Computer Management MMC (Microsoft Management Console) snap-in and are documented in the online help.

- (1) Configure the following services to start automatically:
 - (a) DNS client. Enable this only if you need DNS. Many bastion hosts can operate without DNS configured. A web server that does not make any outbound connections does not need to have DNS configured. The DNS client service caches DNS queries for performance, and does not have to be running, even if you do configure DNS.
 - (b) Event Log. Provides the interface for reading and writing the three Windows 2000 Event Logs.
 - (c) Logical Disk Manager (LDM). Manages locally attached disks.
 - (d) Network Connections. Manages network adapters and network settings.

- (e) Plug and Play. Provides hardware device installation and configuration.
 - (f) Protected Storage. Provides protected storage for sensitive data, such as private keys.
 - (g) Remote Procedure Call (RPC portmapper). RPC allows a program on one system to execute a program on another remote system.
 - (h) *RunAs* Service. Enable this only if you want to be able to use the *runas* command. *runas* is a new feature in Windows 2000 that allows a user to run programs as another user. It is the Windows equivalent of the Unix *su* command. This way, system administrators don't have to be logged in with Administrator privileges. Instead, they can start administrative tools using the *runas* command when they need to perform an administrative task.
 - (i) Security Accounts Manager (SAM). Manages the local user accounts database.
 - (j) Task Scheduler. Schedules tasks for later execution.
 - (k) Windows Management Instrumentation (WMI). This service needs to be started for MMC to work.
 - (l) Windows Management Instrumentation Driver Extensions. This service needs to be started for MMC to work.
- (2) Configure the Logical Disk Manager Administrative Service to start manually.
 - (3) Disable all the remaining services. Sort on Startup Type in the Services view to get a good overview while doing your configuration. Figure 4.2

shows the remaining services running on my own test system (a web server running IIS).



Name	Status	Startup Type	Log On As
Event Log	Started	Automatic	LocalSystem
Logical Disk Manager	Started	Automatic	LocalSystem
Plug and Play	Started	Automatic	LocalSystem
Protected Storage	Started	Automatic	LocalSystem
Remote Procedure Call (RPC)	Started	Automatic	LocalSystem
Security Accounts Manager	Started	Automatic	LocalSystem
Windows Management Instrumentation	Started	Automatic	LocalSystem
Windows Management Instrumentation Driver Extensions	Started	Automatic	LocalSystem
IIS Admin Service	Started	Manual	LocalSystem
Logical Disk Manager Administrative Service	Started	Manual	LocalSystem
World Wide Web Publishing Service	Started	Manual	LocalSystem

Figure 4.2. A List of the Services Running on My Test System.

Disabling NetBIOS

Disable or uninstall the NetBIOS over TCP/IP (nbt.sys) driver in Computer Management → System Tools → Device Manager → View → Show Hidden Devices → Non-Plug and Play Drivers, as shown in Figure 4.3. This disables the Direct Host listener on tcp/445.

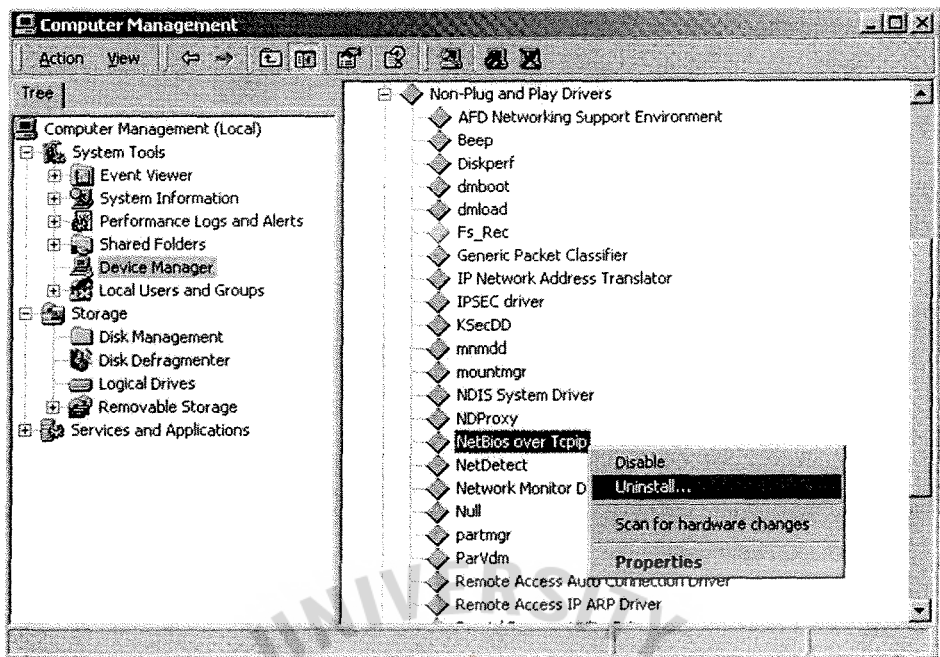


Figure 4.3. The Windows 2000 Device Manager.

Configuring System Processes

Table 4.1 shows the user mode processes that are running, once you have rebooted the Windows 2000 system after disabling the unnecessary services.

Table 4.1. Windows 2000 User Mode Processes Running after Disabling Unnecessary Service.

Process Name	Description
Smss.exe	Session Manager
csrss.exe	Client Server Subsystem
Winlogon.exe	Logon process
services.exe	Main service handler process
Lsass.exe	Local Security Authority and System Accounts Manager server process

Table 4.1. Windows 2000 User Mode Processes Running after Disabling Unnecessary Service (Continued).

Process Name	Description
svchost.exe	Hosts some network services, such as the portmapper (rpcss) and the COM+ event notification system
explorer.exe	Explorer GUI
Winmgmt.exe	Windows Management Instrumentation (WMI)

It is important to understand what is running on your system. The following list describes what each of the remaining processes does:

smss.exe

The Session Manager process is the first user mode process created in the system. During the boot process, it initializes the paging files and loads the kernel Win32 subsystem (*win32k.sys*), among other things. It also starts the Client/Server Subsystem (CSRSS) and Winlogon. After the boot process is completed, the Session Manager main thread monitors the *csrss.exe* and *winlogon.exe* processes. If either of these two processes terminates unexpectedly, the Session Manager will crash the operating system. This is because NT relies on the existence of these processes and cannot function without them.

csrss.exe

The CSRSS process contains the parts of the Win32 subsystem not running in the kernel. These include support for console Win32 applications, error handling, and system shutdown routines. They also provide some supporting Win32 functions—for example, functions for creating and terminating processes.

Winlogon.exe

The Winlogon process is responsible for logging interactive users on to the system. It also ensures access to the trusted path via the secure attention sequence (SAS). The default SAS is the Ctrl-Alt-Delete sequence (the user presses the Delete key while holding the Ctrl and the Alt keys). An application cannot override or intercept the SAS. The existence of the trusted path ensures that it is the operating system, which is a trusted path, that displays the 2000 security dialog box. This means the user is guaranteed that it is the operating system (and not some kind of malicious password-capturing Trojan application) that is asking the user for his password. Winlogon is the process that displays the logon dialog box. It sends the entered username and password to the Local Security Authority (LSA).

lsass.exe

If the username and password provided match the information in the System Account Manager (SAM) database, the Local Security Authority generates an access token for the user and passes it back to Winlogon. Winlogon uses this access token to start a process called *userinit.exe*. User reads the default shell entry (*explorer.exe*) from the registry and starts the shell process. Then User terminates.

The LSA is responsible for the system policies: who should be granted what rights and what events should be audited. It is also responsible for writing to the security event log.

The LSA does not perform the actual security checks on object access. The Security Reference Monitor (SRM) performs the security checks. The SRM is located in the NT Executive.

The relationships between the different user mode and kernel mode processes related to security are shown in Figure 4.4.

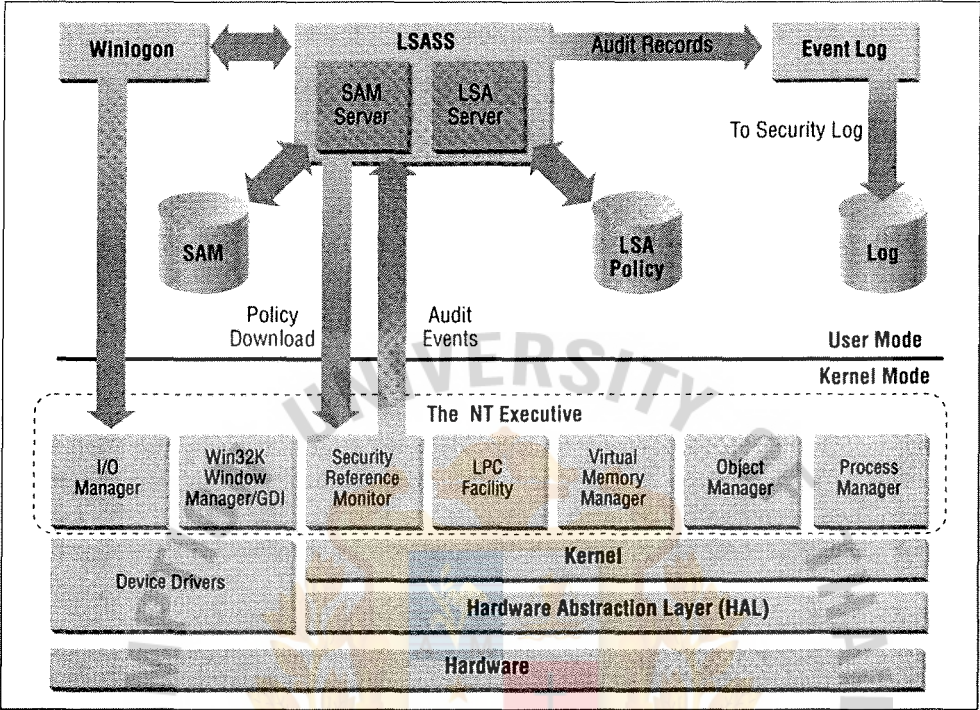


Figure 4.4. The Security-related Processes and Components.

Now that you have removed a lot of unnecessary file- and print-related services, you are left with a pretty lean and manageable system. You also understand the purpose of all active software components on the host.

So far we have created an excellent foundation for building a secure bastion host. However, there is still a lot of room for improvement.

Renaming the Administrator Account

Once you’ve removed the unnecessary services and processes, rename the built-in Administrator account to something unique for your environment. Open the User

Manager and highlight the Administrator account. Use the Rename command on the User menu to rename the Administrator.

This step adds little or no security on a standard 2000 system running the Server service. This is because even if you rename the Administrator account, it still has the same Relative Identifier (RID), also known as the User ID. The RID for the Administrator account is 500. There are tools that can probe a system for the username using unauthenticated “NULL sessions” if you know the RID. See Knowledge Base article Q163846, “SID Values For Default Windows NT Installations,” for a list of all standard RIDs. On a bastion host, however, RID scanning is not possible since the Server service is disabled. For this reason, I recommend renaming the Administrator account.

Make sure there is a strong password on the Administrator account. Use different passwords on all systems. It is good practice to create a separate user account for each person who has access to the system. Grant each user the appropriate privileges in the system. No one should use the Administrator account. Lock away the password for the Administrator account in a safe, and use it only in emergencies.

Configuring the Guest Account

The built-in Guest account cannot be removed. It is disabled by default, and I recommend leaving it that way.

Advanced Configuration

The steps performed so far have used the standard administrative tools in Windows 2000. Now it is time to work on some more advanced tasks to further enhance the security of your system. These tasks include:

- (1) Encrypting the password database
- (2) Editing the registry

(3) Disabling unnecessary files

Protecting the System Accounts Database

If an attacker gets hold of a system backup or an emergency repair disk, he could use a tool such as L0phtCrack to run a dictionary attack or a brute force attack on the Systems Account Manager (SAM) database. However, if the password hashes in the database are encrypted, these attacks will be unsuccessful.

In NT 4.0 Service Pack 3, Microsoft introduced a facility for encrypting the password hashes stored in the SAM database. This facility protects the database from offline password cracking attempts. To implement this encryption feature, run the following command:

```
C:\> syskey
```

Running syskey brings up the dialog box shown in Figure 4.5. Note that enabling password encryption is a one-way operation—once it is enabled, it cannot be disabled.

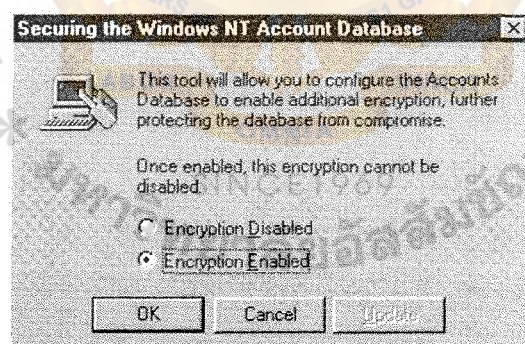


Figure 4.5. The Syskey Command Encryption Dialog Box.

When you enable encryption, the system creates a random 128-bit encryption key. This is used to encrypt the password hash entries in the SAM database in the registry (HKLM\SAM). The encryption key is protected with another key, called the sys tern

key. In the Account Database Key screen (shown in Figure 4.6), choose one of these options:

Password Startup

Specify a startup password here. Later on, this password must be entered at system boot time in order to gain access to the SAM database. The system key is derived from the startup password; it is the startup password's MD5 hash.

System Generated Password

Indicate that the system is to generate a password and choose whether to store the password on a floppy disk or as part of the operating system.

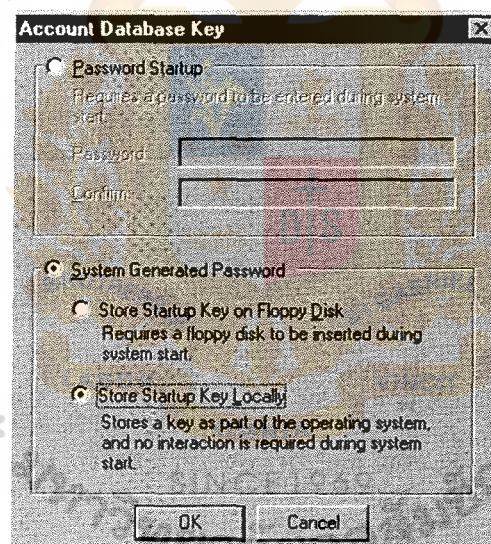


Figure 4.6. The Syskey System Password Options Dialog Box.

Which option should you select? The easiest way to implement SAM database encryption—but also the least secure way—is to use a locally stored system key. The system key is stored on the system disk, using an unspecified obfuscation algorithm. The advantage of using this option is that it does not change the system boot process,

since no user interaction is required to boot the system. The disadvantage is that an attacker may be able to get to the key and decrypt the pass word hashes.

The main advantage of storing the system key on a floppy disk is that it keeps the key off the system disk. However, the floppy disk has to be inserted during sys tem boot, and if the floppy gets corrupted, you cannot boot the system. If you choose this option, be sure to make a copy of the disk right away. Remember to test the copy. Store the original floppy and all copies in a safe manner.

Another option is to provide a startup password when the system boots to decrypt the password key. The main disadvantage is that if you use this option, you won't be able to boot the machine without user intervention. Someone has to physically be at the console to type in the password when the system reboots. The system does not enforce a system key password length, but Microsoft recommends that the startup password be at least 12 characters long. The maximum system key password length is 128 characters.

I recommend using a locally stored system key, a default setting on Windows 2000, for most bastion hosts. The other options introduce more administrative pain than they add security. Microsoft claims that they will support tamper-proof devices (such as smart cards) for system key storage in the future.

Setting System Policies

A policy is a set of rules governing how to use a system. Typically, a security officer decides upon the policy, the administrator implements it, and the operating system enforces it.

In Windows 2000, there are three basic types of policies:

Account policies

These control the characteristics of user accounts. Examples are the minimum length of a password in the system, and how long a user can keep a password before being required to change it.

Audit policies

These control what events will be logged in the system. Examples are logons and logoffs, and file and object accesses.

User rights policies

These control what rights individual users or groups of users have. Examples are the right to access the computer or the ability to back up files.

This section focuses on account policies and user rights policies. Audit policies are discussed in “Auditing and Monitoring Your Perimeter Network” part

Specifying the Account Policy

The account policy controls various characteristics of user accounts on the system—for example, the types of passwords that users can supply. The account policy information is stored in the Systems Account Manager (SAM) database, not in the Local Security Authority (LSA) policy database. To edit the account policy goes to the Local Security Setting → Security Setting → Account Policy.

I recommend the account policy settings listed in Table 4.2.

Table 4.2. Recommended Account Policy.

Policy Setting	Default Setting	Recommended Setting
Password Policy		
Minimum password length	None	8 characters
Enforce password history	Do not keep password history	Remember the last 8 passwords
Minimum password age	Allow changes immediately	Allow changes after 1 day
Maximum password age	Password expires in 42 days	Password expires in 90 days
Password must meet complexity requirements	No	Yes
Account Lockout		
account lockout threshold	No account lockout	5
Reset account lockout counter after	N/A	120 minutes
Account Lockout duration	N/A	1440 minutes (24 hours)

The “Password must meet complexity requirements” option refers to the password which contains the following characteristics:

- (1) Passwords must be at least six characters long.
- (2) Passwords must contain characters from at least three of the following four categories:

- (a) Uppercase letters (A, B, C, ... Z)
 - (b) Lowercase letters (a, b, c, ... z)
 - (c) Numerals (0, 1, 2, ... 9)
 - (d) Nonalphanumeric characters (e.g., @, #, !, and so on)
- (3) Passwords may not contain the username or any part of the user's full name.

Setting Privileges and Rights

You establish a User Rights policy for your bastion host by specifying what individual users and groups of users can do in the system. The term "User Rights" is the friendly name used in the GUI for the operating system rights and privileges.

Rights are used to restrict logon to a Windows 2000 system. The Local Security Authority (LSA) checks that a user has the appropriate right (the right to log on locally, for example) before creating the access token.

Privileges are used to control who can perform certain actions in the operating system—for example, who can shut down the system or change the system time. A user's effective privileges are stored in the access token.

Rights and privileges can be granted to users and groups. A user's effective rights and privileges are the rights and privileges of the user account combined with the rights and privileges of the groups to which the user account belongs.

Assigned privileges and rights are stored in the LSA policy database in the registry v (HKLM\ Security). Privileges and rights are assigned using the Local Security Setting (go to Security Setting→ Local Policies→ User Right Assignment).

The rights and privileges that exist in Windows 2000 are listed in Table 4.3.

Table 4.3. Windows 2000 Rights and Privileges.

User Right	Privilege or Right	Description
Access this computer from the network	Right	Allows a user to access this computer over the network (using the Server service).
Log on as a batch job	Right	Required by COM servers that want to run a class not designed as an 2000 service process in the security context of another user.
Log on locally	Right	Required to be able to log on interactively.
Log on as a service	Right	Required to be able to run as a service process.
Act as part of the operating system	Privilege	A process or thread with this privilege is allowed to act as part of the trusted computing base (i.e., the operating system).
Back up files and directories	Privilege	A user with this privilege can override any NTFS permissions to read a filesystem object. This is required to perform backup operations.

Table 4.3. Windows 2000 Rights and Privileges (Continued).

User Right	Privilege or Right	Description
Bypass traverse checking	Privilege	A user with this privilege can access a subdirectory even though he doesn't have access to its parent directories. This privilege is also required to receive notifications of changes to files or directories.
Change the system time	Privilege	Allows a user to change the system time.
Create a pagefile	Privilege	Allows a user to add or remove a pagefile.
Create a token object	Privilege	Required to create an access token.
Create permanent shared objects	Privilege	Allows creation of special permanent objects, such as <code>\Device\Floppy()</code> .
Debug programs	Privilege	Allows a user to use debugger functions such as <code>OpenProcess()</code> and <code>ReadProcessMemory()</code>
Force shutdown from a remote system	Privilege	A user with this privilege can shut down a computer (via the Server service) remotely.
Generate security audits	Privilege	Required for processes to be able to generate entries in the security log.

Table 4.3. Windows 2000 Rights and Privileges (Continued).

User Right	Privilege or Right	Description
Increase quotas	Privilege	Required to call the <i>CreateProcessAsuser()</i> API function, which creates a new process running in the security context of another user.
Increase scheduling priority	Privilege	Allows a user to change the base priority of a process.
Load and unload device drivers	Privilege	Required in order to load and unload device drivers.
Manage auditing and security log	Privilege	Allows a user to change the audit policy and manage the security log.
Modify firmware environment values	Privilege	Required to change the contents of a computer's NVRAM.
Profile single process	Privilege	Required to access performance counters for a process.
Profile system performance	Privilege	Required to access performance counters for the system.
Replace a process-level token	Privilege	Required to assign the access token of process.

Table 4.3. Windows 2000 Rights and Privileges (Continued).

User Right	Privilege or Right	Description
Create a machine account	Privilege	Allows a user to create a machine account; only valid on a domain controller.
Lock physical pages in memory	Privilege	Allows a process to lock pages in memory. Locked pages cannot be paged to disk.
Restore files and directories	Privilege	Enables a user to set any valid user or group SID as the owner of an object. This is required to perform restore operations.
Shut down the system	Privilege	Allows a local user to shut down the system.
Take ownership of files or other objects	Privilege	Allows a user to take ownership of an object.

On a bastion host system, the *Administrators* group should be granted only the following rights:

- (1) Change the system time
- (2) Create a pagefile
- (3) Load and unload device drivers
- (4) Log on locally
- (5) Manage auditing and security log

- (6) Profile single process
- (7) Profile system performance
- (8) Shut down the system
- (9) Take ownership of files or other objects

The *Backup Operators* group should only have the following rights:

- (1) Back up files and directories
- (2) Log on locally
- (3) Restore files and directories

If you don't want members of the *Backup Operators* group to be able to restore files, revoke the restore right for this group. Create a new group called *Restore Operators* and grant it the following rights:

- (1) Log on locally
- (2) Restore files and directories

The *Users* group should have the right to:

- (1) Log on locally (this is the default)

Don't assign any other rights unless your applications have specific needs. This means that you must revoke all other rights that have been assigned by default.

TCP/IP Configuration

The Internet Protocol (IP) is the language of the Internet. All computers on the Internet exchange data using TCP/IP. The current versions of TCP, UDP, and the IP protocol (Version 4) all became Internet standards in 1981.

An issue with TCP/IP is it offers no form of security such as authentication, encryption, or data integrity. There are many methods of attacking the existing TCP/IP protocol suite. These attacks include:

Connection Hijacking

This occurs when an attacker takes over an existing session. It is, for example, possible for an attacker to take over a Telnet session after a user has logged in. The attacker has to be able to listen to any packets exchanged between the server and the client, and forge packets to launch this attack.

Data Insertion

A stealth attack similar to connection hijacking. The goal is to insert data into an existing session to run commands to break into or to sabotage the target system.

Denial of Service

The purpose of a denial of service attack is to make a site unavailable to normal users. Flooding the network connection of a server with connection attempts may achieve this.

Man-in-the-middle

An attack that tricks a client into believing that it is talking to the real destination server. In fact, it is talking to another system (controlled by the attacker) that is, in turn, talking to the real destination server. This attack is often launched by creating a fake cache entry in a DNS server.

Replay

If an attacker can record a session to a server, he may be able to replay a modified version of this session using a network utility.

Fortunately, the Windows 2000 TCP/IP protocol stack can be hardened to resist some low-level network-based attacks. It is also possible to block unwanted types of network traffic to a host.

Configuring TCP/IP Security Settings

Windows 2000 ships with a basic TCP/IP packet filtering capability known as TCP/IP Security. This software component lets the administrator set a policy that allows or denies certain TCP/IP traffic to a host on a port-by-port basis per network adapter. Figure 4.7 illustrates how TCP/IP Security can be configured to only allow net work access for certain processes listening on allowed ports. All other inbound TCP/IP traffic is denied.

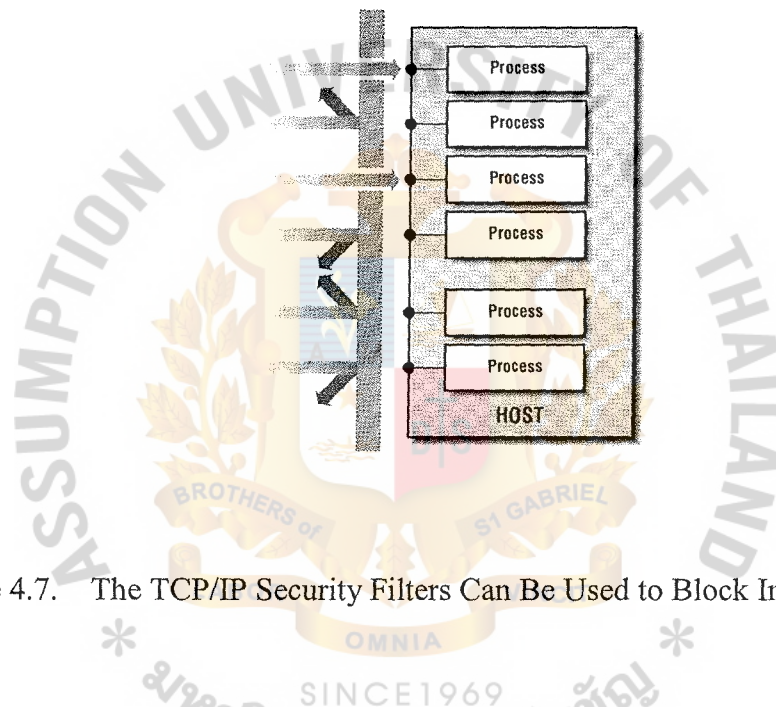


Figure 4.7. The TCP/IP Security Filters Can Be Used to Block Inbound Traffic.

Configure TCP/IP Security by specifying the ports and protocols that are allowed inbound (TCP or UDP) on each network adapter. This is done with the Network application in Control Panel → Network and Dial-up Connections → Local Area Connection (name of your network connection) Properties → TCP/IP → Advanced → Options → TCP/IP Filtering. A sample configuration from a web server is shown in Figure 4.8.

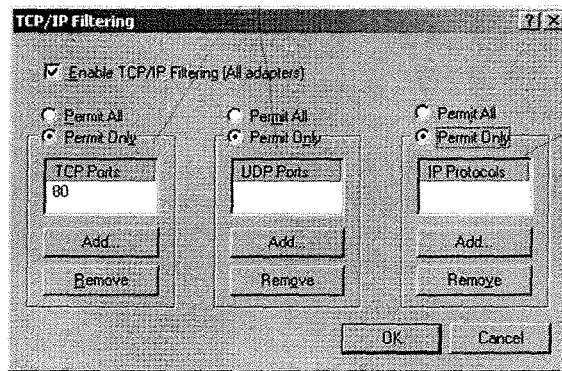


Figure 4.8. The TCP/IP Filtering Dialog Box in Windows 2000.

As you can see, the dialog box lets you control which TCP and UDP ports to allow traffic to. If you allow traffic to a port, it means that TCP/IP filters pass the data to the application listening on that port. If a connection attempt is made to a denied port, the TCP/IP filters discard the connection attempt. The application does not have any knowledge of this. You can also allow other selected protocol by adding them in the “Permit Only IP Protocols” list.

The “Permit Only IP Protocols” list is confusing, to say the least. First of all, it is not possible to block ICMP (IP protocol 1). Second, blocking TCP (protocol 6) or UDP (protocol 17) does not have any effect on this list.

If you define a policy to only allow TCP, the intuitive thing would be to edit the IP protocols list to allow only protocol 6. However, this is wrong since TCP is not controlled by this list. To block everything but TCP (and ICMP), specify these TCP/ IP Security settings:

- (1) Permit All (or some) TCP Ports
- (2) Permit Only UDP Ports: none
- (3) Permit Only IP Protocols: none

Protecting Against SYN Flooding

Denial of service attacks allow a malicious user to crash or consume computing resources on a target system to the point where it can no longer function properly.

The goal of this type of attack is to deny legitimate users access to the service. SYN flooding is by far the most common form of the denial of service attack.

How SYN flooding works

To understand how SYN flooding works, you need to be familiar with how the TCP session setup works:

- (1) The client starts by sending a synchronize sequence numbers (SYN) message to the server.
- (2) The server then answers with a SYN/ACK message. The connection is now half-open.
- (3) The client finishes the handshake with an ACK to the server's SYN. The connection is now fully open and data can be sent in both directions.

This message flow is known as the TCP three-way handshake, and is illustrated in Figure 4.9.

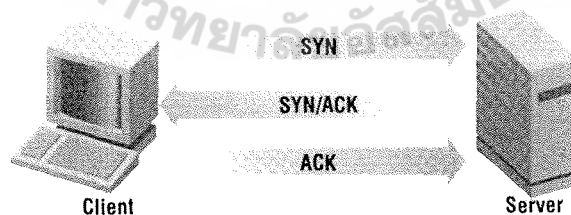


Figure 4.9. The TCP Three-way Handshake.

A SYN flooding attack is accomplished by sending a flood of SYN messages to the server with forged source addresses. Sending a forged source address is commonly

referred to as IP address spoofing. A flood of such messages results in an equal number of half-open connections at the attacked server, since the server will never receive an ACK back on the SYN/ACK messages it sends. As a consequence, the maximum number of half-open connections on the server eventually is reached, and all new TCP connection attempts are denied.

The SYN Attack Protection Feature

It is almost impossible to protect against SYN flooding attacks. There are, however, ways to make your systems better able to resist such attacks. The SYN attack protection feature involves reducing the number of SYN/ACK retransmissions. When enabled (the SynAttackProtect value is set to 1), if the host finds itself under attack, the operating system delays the creation of the route cache entry until a connection is made. Additionally, if SynAttackProtect is set to 2, the operating system will not notify the Winsock kernel mode driver (AFD) until the TCP three-way hand shake is complete

Three settings are used as thresholds to decide if an attack is in progress

TcpMaxHalfOpen

The maximum number of connections in SYN_RCVD state (a SYN has been received, but there's been no ACK response to the outgoing SYN/ACK) before the system will change to SynAttack state.

TcpMaxHalfOpenRetried

The maximum number of connections in a SYN_RCVD for which there has been one or more retransmissions of the outgoing SYN/ACK before the system will change to SynAttack state.

TcpMaxPortsExhausted

When the system has refused the number of connect requests (because the available backlog for connections is at 0), the system changes to SynAttack state.

Recommended settings for the SynAttack protection feature are stated in Table 4.4.

Table 4.4. The SynAttack Setting.

Value Name	Type	Default Value	Recommended Value
HKLM\SYSTEM\CurrentControlSet\ Services\Tcpip\Parameters\ SynAttackProtect	REG_DWORD	0	2
HKLM\SYSTEM\CurrentControlSet\ Services\Tcpip\ Parameters\ TcpMaxHalfOpen	REG_DWORD	100	Default
HKLM\ SYSTEM\urrentControlset\ Servfces\Tcpip\ Parameters\ TcpMaxHalfOpenedRetried	REG_DWORD	80	Default
HKLM\SYSTEM\CurrentControlSet\ Services\ Tcpip\ Parameters\ TcpMaxPortsExhausted	REG_DWORD	5	Default

TCP SYN/ACK Retransmission

Configure your Windows IP stack to be more resilient to SYN flooding by timing out half-open connections more quickly.

By default, the IP stack attempts to retransmit a SYN/ACK message three times. The first retransmission is done after 3 seconds; the second attempt is done 6 seconds after that; and the third attempt is done after an additional 12 seconds. The system waits

24 seconds for an ACK on the last SYN/ACK before giving up. Note that the wait time doubles for each retransmission attempt. As a result, the cleanup occurs after 45 seconds. The setting is shown in Table 4.5.

Table 4.5. The SYN/ACK Setting.

Value Name	Type	Recommended Value
HKLM\System\CurrentControlSet\ Services\Tcpip\Parameters\ TcpMaxConnectResponseRetransmissions	REG_DWORD	3

If you are the target of SYN flooding attacks, set this value to 1 instead to do only one retransmission. As a result, cleanup occurs in only nine seconds. Disable retransmission by setting the value to 0.

Winsock Application Backlog

Windows Sockets (Winsock) is an API for sending and receiving data on a network. Winsock is based on the Berkeley socket interface, which is available on most Unix operating systems.

The Winsock kernel mode driver (*afd.sys*) handles connections for Winsock applications. When a Winsock application calls the *listen()* API function to listen for incoming connections on a socket, it must specify a backlog size. This size is the maximum length of the queue of pending connections. The maximum backlog size is 200 on Windows Server and 5 on Windows Workstation.

A feature called dynamic backlog was introduced in NT 4.0 Service Pack 2. Enabling this feature makes the operating system adjust the size of the backlogs as needed. This provides additional protection against denial of service attacks. It is turned off by default, and is enabled by changing the following registry keys (see the Table 4.6 for specific recommended values):

MinimumDynamicBacklog

The system creates additional free connections for the application when the number of free connections is less than the value specified in this key.

MaximumDynamicBacklog

The system does not create any more connections for an application than the number specified in this key.

DynamicBacklogGrowthDelta

This key controls the number of free connections to create at a time.

Table 4.6. The Dynamic Backlog Setting.

Value Name	Type	Recommended Value
HKLM\ SYSTEM\ CurrentControlSet\Services\ AFD\Parameters\EnableDynamicBacklog	REG_DWORD	1
HKLM\ SYSTEM\ CurrentControlSet\Services\ AFD\Parameters\MinimumDynamicBacklog	REG_DWORD	20
HKLM\ SYSTEM\ CurrentControlSet\ Services\ AFD\ Parameters\MaximumDynamicBacklog	REG_DWORD	5000 ^a
HKLM\ SYSTEM\ CurrentControlSet\ Services\ AFD\Parameters\ DynamicBacklogGrowthDelta	REG_DWORD	10

^a This value is memory-dependent. Setting this parameter to more than 5000 per every 32 MB of RAM in the system leads to memory exhaustion when the system is under attack.

To take advantage of the dynamic backlog feature, applications must request a backlog greater than *MinimumDynamicBacklog* in their *listen()* calls.

Configuring the TCP Keep-Alive Timer

An application can request that TCP keep-alive packets be sent at regular intervals by using the *setsockopt()* system call. This feature is used for keeping an idle connection from terminating. Keep-alives may be used by a news service such as Reuters that pushes new news articles to its clients at different intervals.

To avoid having dead connections consuming resources for an extended period of time, you can lower the connection keep-alive check to every five minutes by setting the registry key as stated in Table 4.7. (the default is every two hours):

Table 4.7. The TCP Keep-alive Timer.

Value Name	Type	Recommended Value (in ms)
HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\KeepAliveTime	REG_DWORD	300000

Configuring Path MTU Discovery

The Windows TCP/IP stack attempts to discover the maximum transmission unit (MTU) for the path to a connecting host. The purpose of this attempt is to eliminate fragmentation at routers along the path. This mechanism can, however, be targeted by a denial of service attack. For example, a malicious user can send spoofed “destination unreachable—packet too big, need to fragment” ICMP datagrams. Sending these datagrams forces the server to start using very small packet sizes, which results in poor performance.

Path MTU discovery is performed by default. For best performance, you should leave this setting on. However, make sure that you allow “need to fragment” messages (ICMP type 3, code 4) to the public servers from external networks.

Path MTU discovery can be disabled by adding the registry key as stated in Table 4.8.

Table 4.8. The MTU Discovery Setting.

Value Name	Type	Recommended Value
HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\EnablePMTUDiscovery	REG_DWORD	0

This setting limits the MTU to 576 bytes for all nonlocal (outside the local subnet) connections, and the operating system does not adjust the MTU on request.

Configuring Source Routing

The routers normally handle routing decisions. Every router matches the destination IP address of a datagram against its routing table in order to decide where to send the datagram next.

This behavior can be overridden by setting the source routing option in the IP header. This allows a client to specify a preferred route for datagrams on a per connection basis. Select source routing by specifying an appropriate value in the IP header.

Source routing can be used for testing, but there is no other legitimate use for source routing. There is no reason whatsoever to allow source routing in perimeter networks. On such networks, source routing could be used by a malicious user to override the normal routing decisions. Therefore, all routers in the network should block IP datagrams with the source routing option set.

Windows servers can also be set up to block source-routed datagrams. Source routing can be disabled by creating the registry entry as stated in Table 4.9.

Table 4.9. The Source Routing Setting.

Value Name	Type	Recommended Value
HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\Disable	REG_DWORD	2

Configuring Dead Gateway Detection

Dead gateway detection is a mechanism to fail-over to the next configured default gateway. TCP informs the IP layer that it thinks the default gateway is down. This occurs after several TCP retransmissions with no response. This feature is susceptible to denial of service attacks. Make the registry change as stated in Table 4.10. to disable dead gateway detection.

Table 4.10. The Dead Gateway Detection Setting.

Value Name	Type	Recommended Value
HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\EnableDeadGwDetect	REG_DWORD	0

Configuring Router Discovery

A Windows host can get a default route by listening for ICMP Router Discovery messages. You need to configure your routers to send ICMP Internet Router Discovery Protocol (IRDP) messages. Note that IRDP may be used by an attacker to masquerade

as the default router. Router discovery is configured on a per-interface basis. By default, the system does not pick up IRDP messages unless configured to do so by DHCP. However, I recommend completely disabling this feature for all network interfaces by changing registry as stated in Table 4.11. If IDRP is not used in your perimeter.

Table 4.11. The Router Discovery Setting.

Value Name	Type	Recommended Value
HKLM\System\CurrentControlSet\Services\ [InterfaceName] \Parameters\Tcpip PerformRouterDiscovery	REG_DWORD	0

Configuring ICMP Redirects

ICMP redirect messages may be used to alter the routing table of a host. This setting is enabled by default and should be disabled on a bastion host. Make the following registry change as stated in Table 4.12 on your system to ignore ICMP redirect messages.

Table 4.12. The ICMP Redirects Setting.

Value Name	Type	Recommended Value
HKLM\System\CurrentControlset\ Services\ Tcpip\Parameters\EnahleICMPRedirect	REG_DWORD	0

Configuring Open Ports

Even though we've removed all NetBIOS-related services from the bastion host at this point, we still have some listeners running. *netstat -an* shows the following output on my own test system:

```
C:\> netstat -an
```

Active Connections

<i>Proto</i>	<i>Local Address</i>	<i>Foreign Address</i>	<i>State</i>
<i>TCP</i>	<i>0.0.0.0:135</i>	<i>0.0.0.0:0</i>	<i>LISTENING</i>
<i>TCP</i>	<i>0.0.0.0:1027</i>	<i>0.0.0.0:0</i>	<i>LISTENING</i>
<i>TCP</i>	<i>127.0.0.1:1025</i>	<i>127.0.0.1:1028</i>	<i>ESTABLISHED</i>
<i>TCP</i>	<i>127.0.0.1:1028</i>	<i>127.0.0.1:1025</i>	<i>ESTABLISHED</i>
<i>UDP</i>	<i>0.0.0.0:135</i>	<i>*.*</i>	

The 127.0.0.1 entries are loopback listeners. These listeners are only accessible from the local computer. They cannot be accessed over the network. The MS RPC portmapper uses tcp/135 and udp/135. This leaves us with two unidentified listeners (tcp/1027 and tcp/1028). The portmapper creates these listeners but they are undocumented.

Although it is possible to make Windows 2000 stop listening on all ports, many applications rely on RPC communication between components on the same computer (especially Microsoft applications). For example, Internet Information Server 4.0 breaks if you disable the RPC client or server. However, if you do not need RPC on your system, disable it by removing the keys as listed in Table 4.13 in the registry:

Table 4.13. The RPC Setting.

Value Name	Type	Recommended Value
HKLM\Software\Microsoft\RPC\ ClientProtocols\ncacn_ip_tcp	REG_SZ	Remove
HLCLM\Software\Microsoft\RPC\ ClientProtocols\ncagd_ip_udp	REG_SZ	Remove
HKLM\ Software\Microsoft\RPC\DCOM Protocols	REG_MULTI_SZ	Remove only “ncacn_ip_tcp”

The keys listed are some of the available transport mechanisms for RPC calls. By removing them, we effectively prevent RPC from using TCP or UDP as transports. This leaves you with no open ports whatsoever on your bastion host, as you see from this example:

C:\> netstat -an

Active Connections

Proto Local Address Foreign Address State

C:\>

Configuring Administrative Tools and Utilities

Many system administrators have a set of favorite tools and scripts they copy to the hosts they administer, if you have some tools that you just cannot live without, copy them to a directory of their own (such as *c:\admintools*), create a new group called *Admintools*, and allow only members of that group access to this directory. Consider

encrypting the directory with PGP-disk or a similar tool. Since Local System is a member of the Administrators group, it is important that you do not add the built-in Administrators group to any new groups you create.

You also need to reset permissions on the standard Windows 2000 administrative tools in the system32 directory. I recommend setting the DACL on the files listed in Table 4.14 to Admintools:F (Full Control) only.

Table 4.14. The Administrative Tool and Utility Files.

Filename	Description
arp.exe	Used to display and modify the ARP cache
at .exe	Used to submit jobs to the Schedule service
cacls.exe	Used to modify DACLs on files
cscript.exe	Windows Script Host (if installed)
cmd.exe	Command interpreter
ipconfig.exe	Shows the IP configuration and can be used to manage DHCP addresses
net.exe	Used to manage users, groups, shares, and more
net1.exe	Used to manage users, groups, shares, and more
netstat.exe	Shows network connections, routes, and statistics
Nslookup.exe	DNS lookup
Ntbackup.exe	NT Backup software
ping.exe	Used to test network connectivity using ICMP echo messages
rdisk.exe	Recovery disk utility
regedit.exe	Win9x-style registry editor

Table 4.14. The Administrative Tool and Utility Files (Continued).

Filename	Description
regedt32.exe	Registry editor
route .exe	Manages routing information
runonce.exe	Adds a program to the RunOnce key used by the Explorer
syskey.exe	Encrypts the SAM database password hashes
tracert.exe	Trace route utility
winmsd.exe	System Information utility
wscript.exe	Windows Script Host (if installed)
xcopy.exe	File copy program

Setting Permissions

Setting permissions on operating system objects such as files, directories, and registry keys provides a fine-grained access control mechanism. In Windows 2000, access to objects is controlled by Discretionary Access Control Lists (DACLS). Each object in the operating system includes a DACL. Consider the following example.

Example: A Sample DACL

Charoenchai : No Access

Administrator : Write, Execute

Users: Read

The DACL shown in this example grants any member of the Administrators group Write and Execute permission. Members of the Users group have Read access. Permissions are cumulative. If a user is a member of both Users and Administrators, his

effective access will be Read, Write, and Execute (the combined permissions of his user and the groups of which he is a member). The user charoenchai's effective permission is "No Access," regardless of which groups of which he is a member. This is because No Access overrides all other permissions.

Setting File-Level Permissions

The Windows NT File System (NTFS) supports the permissions shown in Table 4.15.

Table 4.15. The NTFS Permission.

NTFS Permission	File	Folder
Read (R)	Display the contents of a file and other data such as the owner and permissions.	Display the contents of a folder and other data such as the owner and permissions.
Write (W)	Modify the file.	Add files and folders to the folder.
Execute (X)	Run the file, if it is an executable.	Make changes to folders within the folder.
Delete (D)	Delete the file.	Delete the folder.
Change permissions (P)	Change the file's permissions.	Change the folder's permissions.
Take ownership(0)	Take ownership of the file.	Take ownership of the folder.

There are four predefined groups of privileges for file access control as stated in Table 4.16.

Table 4.16. The Predefined Groups' Permission Setting for File Access.

Group	Permissions
Full Control	RWXDPO
Change	RWXD
Read	RX
No Access	None

There are also seven predefined groups of privileges for directory access control as stated in Table 4.17.

Table 4.17. The Predefined Groups' Permission Setting for Directory Access.

Group	Right Granted to Directory	Permissions on Files
Full Control	RWXDPO	RWXDPO
Change	RWXD	RWXD
Add and Read	RWX	RX
Add	WX	N/A
Read	RX	RX
List	RX	N/A
No Access	None	None

The short list of NTFS DACLs presented throughout this part provides a simple way of ensuring that the Local System account will not be able to execute command line tools that might help an attacker who has gained Local System access to the host.

Setting Registry Permissions

The Windows 2000 registry contains all operating system settings. Obviously, it is important to make sure the registry is properly secured. Much like an NTFS file system, the registry database can also be protected by DACLs. These permissions, however, are not the same.

The permissions available in the registry are shown in the following table.

Table 4.18. The Registry Permission Table.

Registry Permission	Right
Query value (Q)	Read a value
Set value (S)	Write a value
Create subkey (C)	Create a subkey
Enumerate subkeys (F)	List subkeys
Notify (N)	Allow change notification
Create link (L)	Create a symbolic link
Delete (D)	Delete a key or a value
Write DAC (P)	Change permissions
Write owner (O)	Take ownership
Read control (R)	Read security information

The Security Templates and Security Configuration Analysis described in Chapter 3 comes with a template called “hisecws” (highly secured workstation) that you can use for file level and registry permissions on a bastion host (with the exception of the changes suggested in this part). Note that you need to make sure any applications you install have solid permissions set in the filesystem and on their registry keys.

4.2 Setting Up Secure Remote Administration

If you’ve carefully followed the installation and configuration steps outlined in earlier part, your bastion host is now quite secure. However, there is no way of administering it remotely!

Windows 2000’s standard remote administration tools such as Event Viewer and Server Manager are based on RPC using NetBIOS. The problem with NetBIOS is that it is considered unacceptably insecure in perimeter networks. Hence, we must find alternative tools for administering and monitoring the Windows 2000 host.

The basic requirements for any new remote management tools are:

Authentication

Both user and source IP address authentication are required to restrict unauthorized access to the servers. IP address authentication means access to certain services can be allowed or rejected based on the IP address of the client.

Encryption

Since we are performing administrative tasks, such as adding users and changing passwords, all sessions must be encrypted.

Ability to use the Windows NT GUI tools

It is not possible to perform all administration tasks using the command prompt. Hence, we need some kind of remote graphics console.

Ability to transfer files from and to the remote system

It is often important to be able to transfer files to your bastion hosts in a simple way.

This part presents a solution for remote management of Windows 2000 servers:

Windows 2000 Terminal Services

A remote management solution that is available on Windows 2000 Servers only (not Windows 2000 Professional).

This part walks you through the necessary steps to successfully install and configure this one.

4.2.1 Windows 2000 Terminal Services

If you run a site with Windows 2000 systems, you might want to use the built-in Terminal Services (TS) in Windows 2000 (Server versions only) for remote administration.

Terminal Services is based on Microsoft's Remote Desktop Protocol (RDP). RDP is a Microsoft proprietary protocol. Terminal Services does not provide built-in support for file transfer. RDP uses only one port (tcp/3389), which is good. Also, TS remote administration supports two concurrent remote users.

Configuring Terminal Services for Remote Administration

Follow these steps to set up Terminal Services on a bastion host:

- (1) Install the Terminal Services component by clicking Add/Remove Programs in the Control Panel, and then clicking Add/Remove Windows Components.

There's no need to install the Terminal Services Licensing service when using the remote administration mode. The Windows Components dialog box is shown in Figure 4.10.

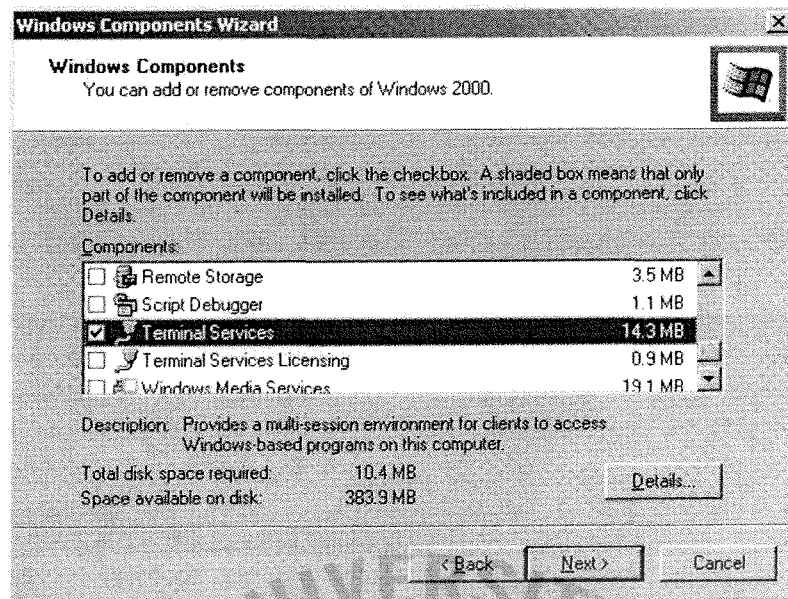


Figure 4.10. The Windows Components Dialog Box.

- (2) Configure Terminal Services to use the remote administration mode. Terminal Services can run in either remote administration mode or application server mode. To set up a dedicated application server with Terminal Services, separate licenses are needed. The remote administration feature is included in the Windows 2000 Server license. In this case, I choose to configure the Terminal Services for remote administration as shown in Figure 4.11.

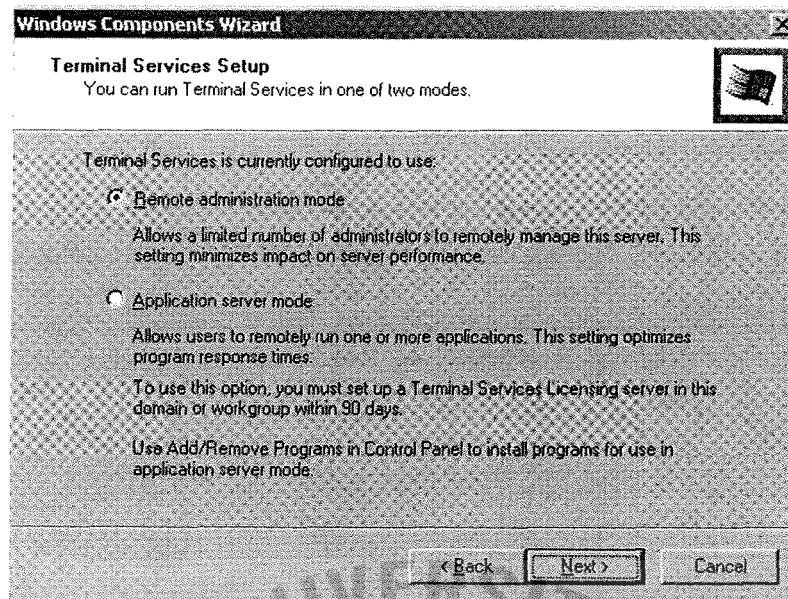


Figure 4.11. Terminal Services Mode of Operation Configuration.

- (3) Remove the TslnternetUser account from the system. Terminal Services creates a local user account on the system called "TslnternetUser" as a part of the installation procedure. This account is used for the Terminal Server Internet Connector licensing mode (anonymous Internet access to the Terminal Service). The account is not needed for remote administration, and therefore I recommend removing this account from the bastion host.

The next major task is to configure Terminal Services. This can be done by using the Terminal Services Configuration MMC snap-in available in the Programs → Administrative Tools folder on the Start menu. Bring up the RDP-Tcp Connection Method Properties, as shown in Figure 4.12.

The settings we need to configure include:

- (1) The level of encryption for connections to the Terminal Service
- (2) Terminal Services session settings

- (3) Permissions to control who is allowed to access the server using Terminal Services

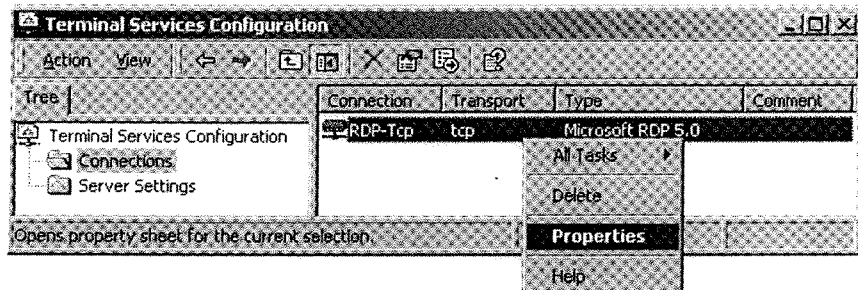


Figure 4.12. The Terminal Services Configuration MMC Snap-in.

RDP uses the RC4 cipher using 40-bit, 56-bit, and 128-bit encryption keys to protect against eavesdropping on Terminal Services connections. RDP supports three different methods of encryption:

Low

Encrypts only input sent from the client to the server (like username and password information). Do not use this setting on a bastion host.

Medium

Encrypts all data sent between the server and the client using either a 56-bit key (Windows 2000 TS clients) or a 40-bit key (older TS clients).

High

Encrypts all data sent between the server and the client, using a 128-bit key. The Windows 2000 High Encryption Pack must be installed on both clients and servers to get 128-bit encryption. This setting will fall back to 56-bit encryption if either the client or the server doesn't have high encryption installed.

Configure Terminal Services according to these steps:

- (1) Configure Terminal Services to use the 128-bit strong (called “high” here) encryption option, as shown in Figure 4.13.

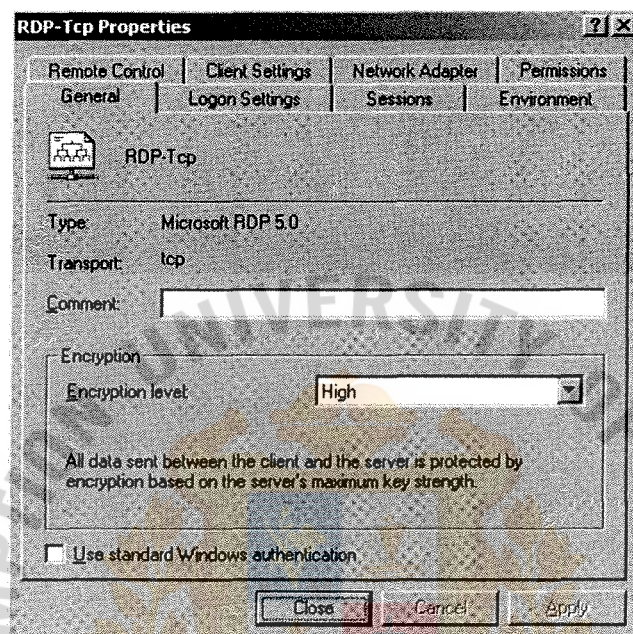


Figure 4.13. RDP Encryption Settings.

- (2) Terminal Services can be configured to disconnect idle connections and to terminate broken sessions. Since the remote administration mode only allows two concurrent sessions, you must make sure that hanging or idle connections are disconnected as soon as possible.

I recommend the settings shown in Table 4.19 on a bastion host.

Table 4.19. Recommended Session Setting for Terminal Services.

Setting	Description	Recommended Value
End a disconnected	A session is considered disconnected if session the user closes the TS client application without logging out first.	30 minutes
Active session limit	Controls how long a user's session can remain active.	Never (no limit)
Idle session limit	Controls how long a user's session can remain idle before it is closed by Terminal Services	10 minutes
When session limit is reached or connection is broken	Setting this to "End session" terminates reached or connection the user's active processes and logs the is broken user out if the session is broken or if one of the above limits is reached.	End session

Configure the settings from Table 4.19 using the Sessions tab (shown in Figure 4.14) of the RDP Properties dialog box.

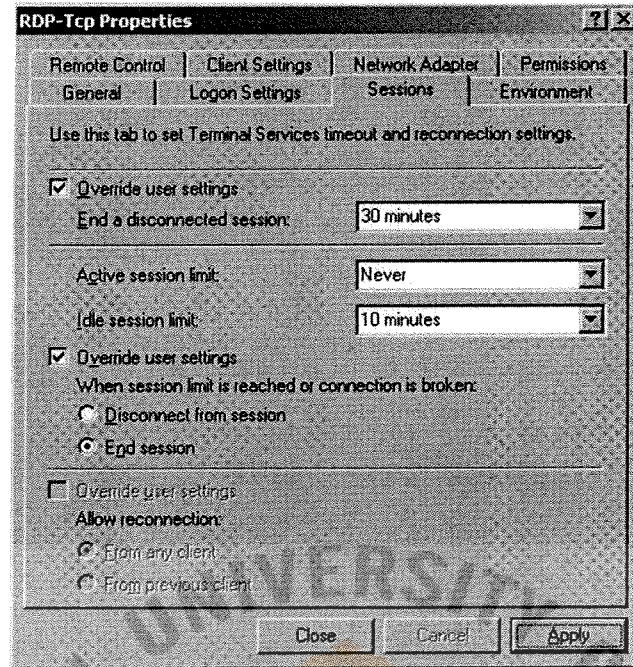


Figure 4.14. RDP Sessions Settings Tab.

- (3) The last configuration step is to set up access control to Terminal Services. This is done using the Permissions tab (shown in Figure 4.15) in the RDP Properties dialog box. By default, all members of the Administrators group are allowed access to Terminal Services. I recommend removing the Administrators group from the allowed users and adding the individual users who need access back instead. Note that the SYSTEM account needs to be in the list. You will not be able to log on if you remove it.

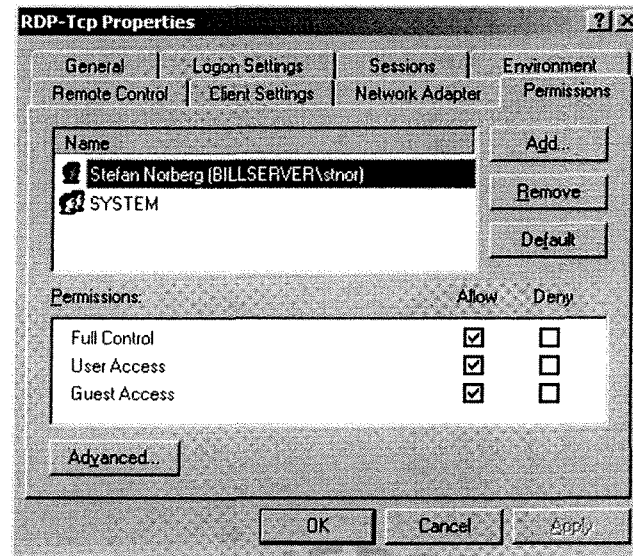


Figure 4.15. The RDP Permissions Tab.

Copying Files over RDP

By default, there is no way to copy files between the client and the server using Terminal Services. However, installing the File Copy utility from the Windows 2000 Server Resource Kit (<ftp://ftp.microsoft.com/reskit/win2000/rdpclip.zip>) will enable you to do this. The File Copy utility provides the ability to use the Cut (Ctrl-x), Copy (Ctrl-c), and Paste (Ctrl-v) clipboard commands to transfer files between the client and the server. Simply select the files and/or folders to copy in Explorer (on the client) and press Ctrl-c. Now all that is left is to go to the Terminal Services Client application and paste the file in a folder of your choice. The file is then transmitted over the RDP channel to the server. Unfortunately, as with all Resource Kit utilities, this great add-on feature is provided as is. It is not supported by Microsoft.

4.3 Backing Up and Restoring Your Bastion Host

Backups are a very important part of any computer operation. Do not neglect them. This part discusses the special procedures to follow when backing up a bastion host. It also provides an introduction to the built-in backup software Windows 2000.

4.3.1 Defining Your Backup Policy

Your organization probably already has a backup policy in place that defines how backups and restores should be handled. Such a backup policy should at least contain answers to the following questions:

- (1) Who is responsible for making backups of your data?
- (2) How often should backups be performed?
- (3) Where should the backup media be stored?
- (4) Who is authorized to restore data to a system?

If you already have a general backup policy for your site, you must decide whether it fits the needs of the bastion hosts in the perimeter network. If the general policy does not fit, you need to define a special backup policy for the systems in the perimeter network.

4.3.2 Backup Methods

Backups can be performed using either a local storage device or a remote storage device. The backup method you choose depends upon your needs and your environment. In a small environment, local backups might be the best solution. If you have a larger number of hosts, you may prefer to use remote backups to centralize management and data storage.

Local Backups

A local backup is a backup performed to a locally attached storage device. Local backups are the preferred way of doing backups in a perimeter network, since this type

doesn't affect the network exposure of the host. No backup agent software that opens up network listeners needs to be installed.

However, there are several problems with doing local backups. One problem is that a local backup architecture might not scale from a management perspective. Since there is no single tool to handle all backups, each host must be administered and monitored separately.

Another problem is that the backup media is left in the backup device until someone manually removes it and takes it to a safe place. There's no point in doing backups in the first place if there's a fire in the computer room and the tape is destroyed.

Backups over the Network

If you have lots of hosts in the perimeter, you may find that it is better to perform backups over the network. A common way to achieve this is to install a backup agent on the bastion hosts that authenticates the backup server and encrypts the data in transit. The backups are initiated by a dedicated backup server that has the backup device(s) attached locally.

Because your backups will inevitably contain some very sensitive data, it is good practice to place the backup server in a separate room, where only a handful of people have access. Especially sensitive data includes the system configuration, possibly some unencrypted passwords, and, of course, the application data.

It can be very challenging to make network backup software work in a firewall environment. Usually the backup server sends a command to start a backup to a "backup agent" (the client software) running on the system that needs to be backed up. The agent then initiates one or several data connections back to the backup server for the actual data transfer. From a network security point of view, it is better if the backup server initiates all connections. That way, you don't have to expose the backup server to

any incoming connections. I recommend placing your backup servers on a separate network in the perimeter, as shown in Figure 4.16.

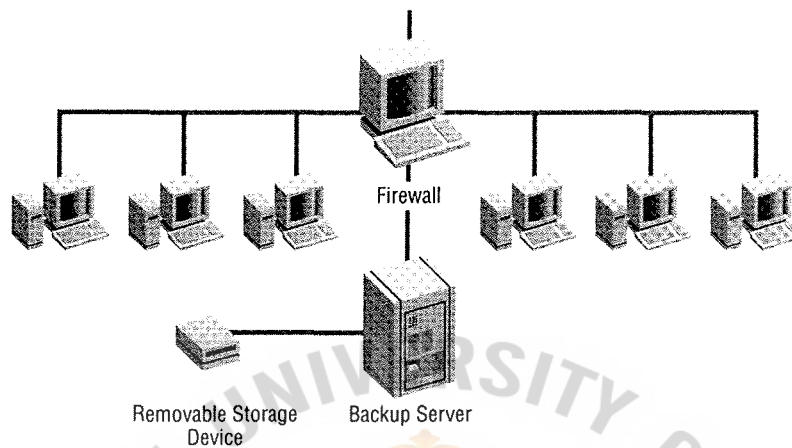


Figure 4.16. A Common Network Backup Design in Perimeter Networks.

Sometimes network backups are performed over a dedicated network to avoid choking the production networks during backup. If you decide to implement a dedicated backup network, it is very important to design this network as you would any other perimeter network segment. In implementing the dedicated backup network, make sure that you don't short-circuit any other security measures you've taken in the perimeter.

The design of the backup network in Figure 4.17, for example, short-circuits firewall. If an attacker gets control over one of the servers, he is able to attack the hosts in the other security compartment using the backup network to bypass the firewall.

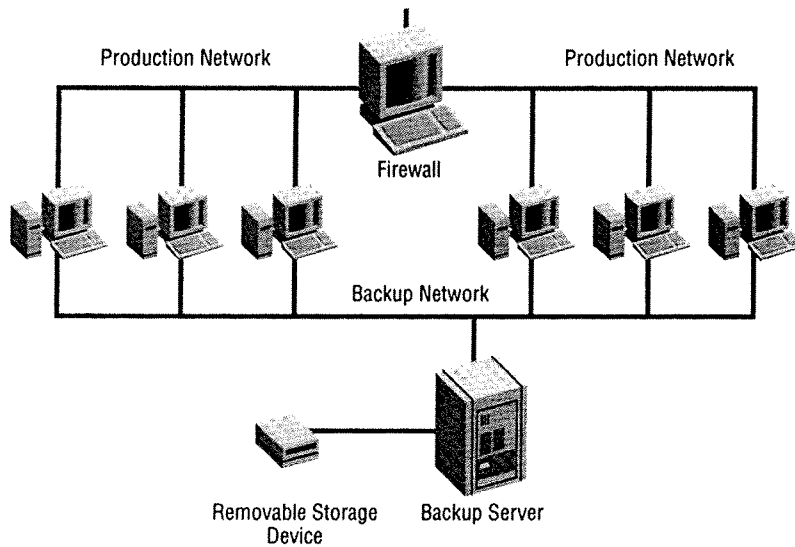


Figure 4.17. An Example of a Poor Backup Network Design.

A better design is to have a separate firewall to achieve a compartmentalized backup network, as shown in Figure 4.18.

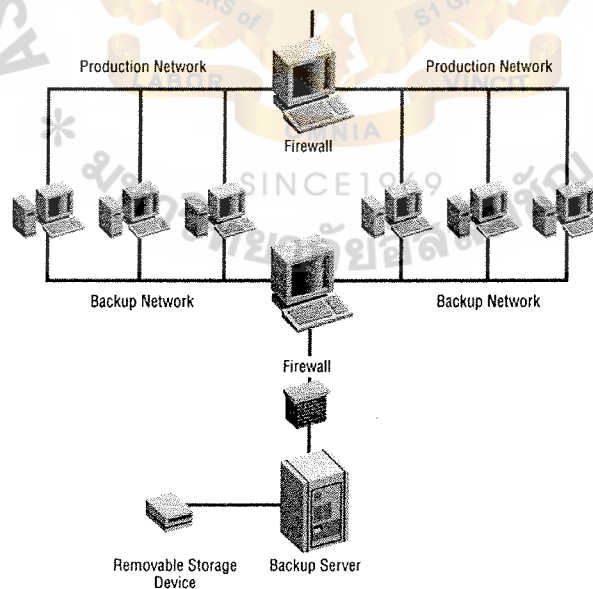


Figure 4.18. An Example of a Good Backup Network Design.

4.3.3 Types of Backups

It is important to back up data frequently. Decide what kind of backup schedule is suitable for your environment. Before deciding what's best for your site, consider what's more important: to have a simple and fast restore procedure, or to perform backups quickly.

Making a Golden Image

The first thing to do is make a golden image of your system. A golden image is a full system backup made before the system is connected to the public network. This image can be used for recovery, but it can also be used as a reference if the system is compromised in any way.

Performing a Full System Backup

A full system backup is a backup of everything on a system: the operating system, the application executables, and the application data. Always perform a full system backup both before and after any major change takes place—for example, when applying a new service pack.

If the main objective of your backup strategy is to restore a system as quickly as possible, you need to schedule full system backups as often as possible.

Performing Incremental Backup

If you have a minimal backup window, you will not have time to make full backups of your systems every day. A common method then is to make a full system backup on Sundays and incremental backups on the other days of the week. An incremental backup contains only the files that have changed since the last backup.

The trade-off with this type of backup is that it makes the restore procedure slower and more complex. If there is a system failure on a Friday, you need to start by

restoring the last full system backup (which was on Sunday), and then all incremental backups (which took place Monday through Thursday).

Performing Differential Backup

The differential backup scheme is a compromise between the full and the incremental schemes. Make a full system backup on Sundays and differential backups on the other days of the week. A differential backup contains the files that changed since the last full system backup. As a result, the differential backup takes longer and longer to run until you perform a new full system backup.

This scheme ensures that only two restore sessions are needed (the full system backup and the last differential backup) to recover a system from failure.

4.3.4 Backup Software

There are many types of products for backup and restore operations. If you have a small environment and are performing local backups, the most obvious choice is to use the backup software included with Windows. Microsoft ships a “light” version of Veritas’ Backup Exec with both Windows NT and Windows 2000. This is called NT Backup. NT Backup can only be used to perform local backups. I briefly describe this software in the next section; however, I recommend instead choosing a full-featured backup software product with features such as disaster recovery.

If you are doing remote backups and restores over the network, there are many choices. Available products include:

Computer Associates’ ArcServe

Software for backing up mainly Netware and NT systems.

Legato Networker

For heterogeneous networks that contain a mix of NT, Unix, and other systems.

Veritas' Backup Exec

An excellent product for backing up and restoring Windows NT/Windows 2000 hosts.

Veritas' Netbackup

Another piece of software for large heterogeneous networks.

When deciding on which backup software to use for remote backups, consider the following:

Is the product suitable for use in a firewall environment?

Does the product operate on dynamically assigned or fixed ports? How many ports are required?

What kind of authentication is used?

Are passwords transmitted in cleartext? Is the authentication protocol documented and/or based on open standards?

Does the product support encryption?

Does the product support encryption between the backup server and the client? Does the product support encryption when writing to the media? Is the encryption based on vendor proprietary or published algorithms?

Does the product support disaster recovery?

Is it possible to perform restoring without having to reinstall and reconfigure hardware and software components?

Windows 2000 Backup

Scheduling backups is more convenient with Windows 2000 than it is with Windows NT. You can easily perform a scheduled backup, as shown in Figure 4.19, because the Windows Backup product is integrated with the Windows 2000 Task Scheduler.

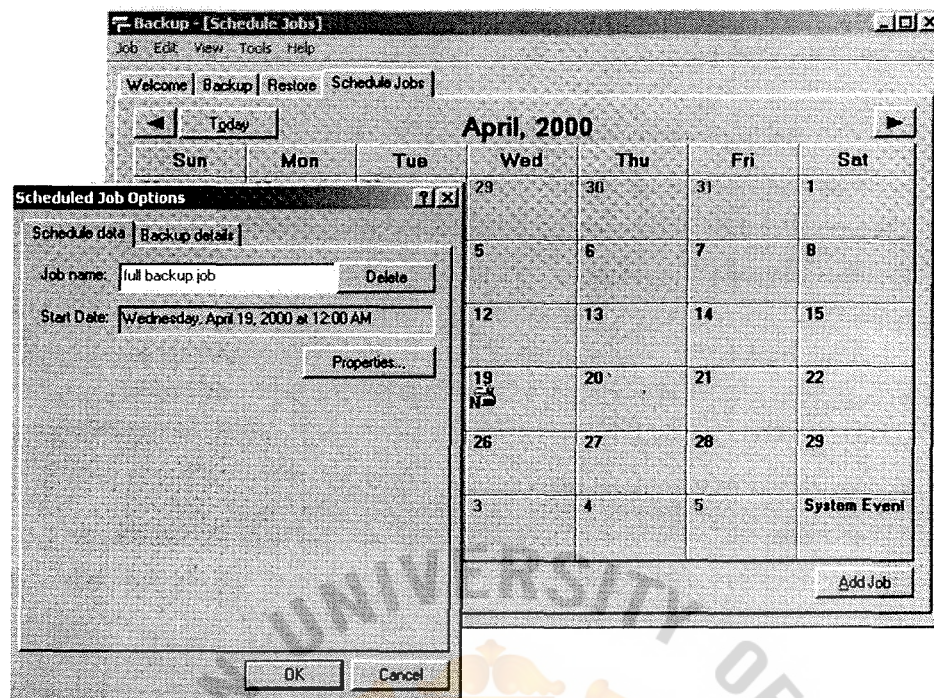


Figure 4.19. Windows Backup (Windows 2000).

There's a helpful article on the Microsoft Technet web site called "Windows 2000 Server Disaster Recovery Guidelines" (<http://www.microsoft.com/technet/win2000/recovery.asp>) that describes how to perform disaster recovery using Windows 2000 Backup.

4.4 Auditing and Monitoring Your Perimeter Network

Systems in a perimeter network need constant monitoring. It is crucial to detect abnormal behavior promptly, since such behavior might indicate a break-in or maybe just an unintentional configuration change that weakens the security of the system. This part discusses how to implement a strong system monitoring solution on Windows. This includes configuring the standard Windows 2000 event logging and auditing systems.

System Auditing in Windows

The auditing system in Windows is very good. It is possible to audit every type of object access in a granular way. An object in Windows NT is anything from a filesystem object (e.g., file or directory) to a printer, registry key, or internal operating system data structure. For instance, it is possible to set up auditing for a single action (e.g., read or write) on a single file for a certain user. System Access Control Lists (SACLs) control how an object is audited.

The Security Reference Monitor (SRM) is responsible for generating audit information based on the SACLs and the audit policy. The auditing information is passed on to the Event Log system, which writes the events to the event log files. Auditing is turned off by default.

The Event Logs

The Event Log service is responsible for writing events that are reported through the Event Log API to the event logs.

There are five types of events:

Information

An event that doesn't indicate an error condition, such as when an application starts successfully.

Warning

An event that doesn't require immediate action, but may result in a more serious condition later on, such as when disk space is low.

Error

An event that indicates an error that has a serious impact on a subsystem or application, such as when an application fails to load,

Audit Success

An event that indicates a successful security check, such as when a user tries to access a file to which he has privilege.

Audit Failure

An event that indicates a failed security check, such as when a user fails to log on.

There are three types of event logs in a Windows 2000 system:

Application log

The Application log (appevent.log) is where applications report their event messages.

System log

The System log (sysevent.log) contains messages from the base operating system components. No applications are allowed to log information here.

Security log

The Security log (secevent.log) contains auditing information.

The Application and System logs contain Information, Warning, and Error events.

The Security log contains Audit Success and Audit Failure events.

The Log Files

The log files are located in %SystemRoot%\system32\config\ by default. Change the location of the logs by specifying new values for the registry keys shown in Table 4.20.

Table 4.20. Log Files Setting.

Value Name	Type	Recommended Value
HKEY_LOCAL_MACHINE\SYSTEM\ CurrentControlSet\ services\EventLog\Application	REG_SZ	File path
HKEY_LOCAL_MACHINE\SYSTEM\ CurrentControlSet\ Services\ EventLog\System	REG_SZ	File path
HKEY_LOCAL_MACHINE\SYSTEM\ CurrentControlSet\ Services\ EventLog\Security	REG_SZ	File path

I recommend using a separate disk or partition for the event log files and other types of log files to avoid creating fragmentation or I/O bottlenecks on other disks. This also makes the Event Log system more resilient to denial of service attacks, assuming that you dedicate a large partition for logging. You must reboot your system for the changes to take effect, since you can not restart the Event Log service manually.

The actual log messages are not written in cleartext, but in a coded binary format. For each event, the Event Log service writes an event ID to the log file, rather than the whole message string, to preserve disk space. The strings needed to format the messages are stored in the event source applications' DLL files. As a result, the log files can't be viewed directly, but only through the Event Log APIs.

The application used to view the event logs is the Event Viewer. The Event Viewer can also be used to change how the Event Log system handles the different log files.

You can set the maximum size of each log and control what happens if a log file fills up. I recommend using a circular logging strategy, which allows the Event Log system to overwrite old events as needed. Increase the maximum log size to 102400 KB (about a hundred megabytes). This setting makes it very time consuming for an attacker to cover his tracks, while keeping the system administrative effort to a minimum; there's no need to empty the log files by hand. Note that this setting is log-specific—you need to reconfigure each log separately.

Security Audit Failure

A common way attackers cover their tracks is by flooding the logs with bogus information. As the number of log entries grows, the log file or filesystem runs out of free space. When the log file is full, the attacker makes his move; at this point, his activities can't be logged because the log file is filled up.

To ensure that all actions in your system are logged, configure Windows to crash if a write to the Security (audit) log fails (because the log is full). Note, however, that an attacker may be able to crash your system by filling up the event log file if you enable this setting. Make the following change to the registry as stated in Table 4.21 to enable this feature:

Table 4.21. Security Audit Failure Setting.

Value Name	Type	Recommended Value
HKEY_LOCAL_MACHINE\SYSTEM\System\CurrentControlSet\Control\Lsa\CrashOnAuditFail	REG_DWORD	1

Only a member of the Administrators group can log on locally after a CrashOnAuditFail.

Additional Registry Settings

There are two registry entries that control additional auditing behavior:

AuditBaseObjects

This setting enables auditing for the so-called Windows base objects. These are internal Windows objects invisible to the user (e.g., the KnownDLLs data structure, and device names such as \Device\CDRom0).

FullPrivilegeAuditing

This setting enables full auditing on backup and restore operations.

Enabling these settings produces very large amounts of data, and I recommend against doing so.

Recommended settings for these entries are shown in Table 4.22.

Table 4.22. Additional Registry Setting.

Value Name	Type	Recommended Value
HKEY_LOCAL_MACHINE\SYSTEM\System\CurrentControlSet\Control\Lsa\AuditBaseObjects	REG_DWORD	0 (default)
HKEY_LOCAL_MACHINE\SYSTEM\System\CurrentControlSet\Control\Lsa\FullPrivilegeAuditing	REG_DWORD	0 (default)

Configuring Auditing

The audit policy in Windows controls which events should be audited. In Windows 2000, it is controlled from the Group Policy snap-in MMC, as shown in Figure 4.20.

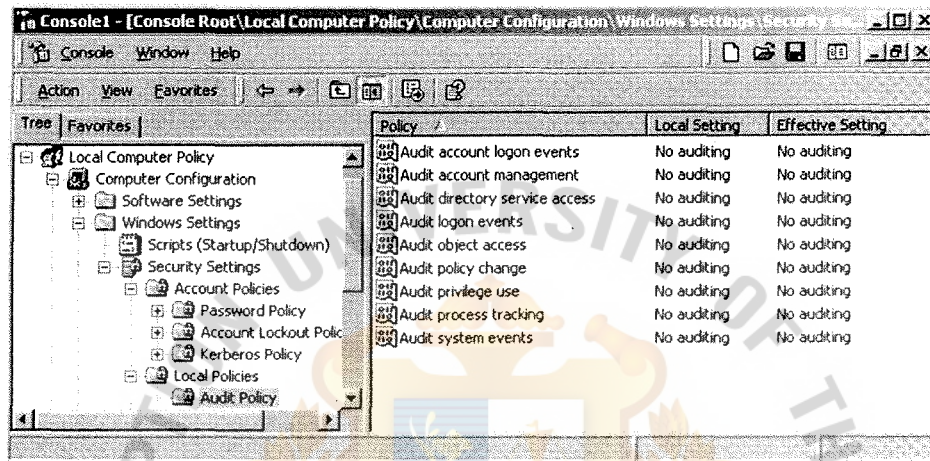


Figure 4.20. Audit Policy Settings in Windows 2000.

There are seven different categories of events on which to control auditing. Table 4.22 explains these categories and the recommended settings. Note that auditing is turned off by default. Right-click on "Security Settings" in the left pane, and choose "Reload" to update the "Effective Setting" shown in the GUI.

Table 4.23. The Audit Policy Settings.

Type of Event	Success	Failure	Recommended Setting
Logon and Logoff	Successful logon or logoff attempts will be audited.	Failed logon attempts will be audited.	Success and Failure. Logon attempts are important events that must be audited.
File and Object Access	Files and other objects with matching SACLs will be audited on success (permission granted).	Files and other objects with matching SACLs will be audited on failure (permission denied).	Success and Failure. File and object access auditing should be enabled in the audit policy, so that it is possible to configure auditing for files, directories, and other system objects using SACLs.

Table 4.23. The Audit Policy Settings (Continued).

Type of Event	Success	Failure	Recommended Setting
Use of User Rights	Successful use of User Rights, such as changing the system time while having the SeSystemtime-Privilege.	Failed use of User Rights, such as trying to debug a program (using the Debugging API) without having the SeDebugPrivilege.	None. Auditing of Use of User Rights” can produce enormous amounts of audit data.
User and Group Management	Successful attempts to modify the accounts database.	Failed attempts to modify the accounts database.	Success and Failure. Modification attempts of the accounts database are important auditing data.
Security Policy Changes	Successful changes of the audit policy.	Failed attempts to change the audit policy.	Success and Failure. We need to monitor changes to the policy itself.

Table 4.23. The Audit Policy Settings (Continued).

Type of Event	Success	Failure	Recommended Setting
Restart, Shutdown, and System	Successful restart, shutdown, or other system-wide security events.	Failed attempts to restart, shut down or perform other system-wide security events.	Success and Failure.
Process Tracking	Successful creation or termination of processes.	Failed attempts to create or terminate processes.	None. Produces too much auditing data to be useful in normal production environments, and degrades system performance

File and Object access Auditing

The Security Reference Monitor in the kernel is responsible for object auditing. File and object access auditing is controlled by object SACs.

To enable auditing on an object like a file or printer, you must first enable auditing of these events in the audit policy. Next, configure auditing for sensitive files and directories. These can include:

- (1) Operating system files such as %SystemRoot%\system32
- (2) Application binaries
- (3) Data directories such as HTML files on a web server

Another good strategy is to perform exception auditing. An example of this is to audit all activity for the anonymous us users IUSR_COMPUTERNAME and IWAM_COMPUTERNAME outside the web root. If these users try to access any files out side the web root, it could indicate an attempt to break into the system.

Filesystem auditing is configured using each object's Properties dialog box, as shown in Figure 4.21.

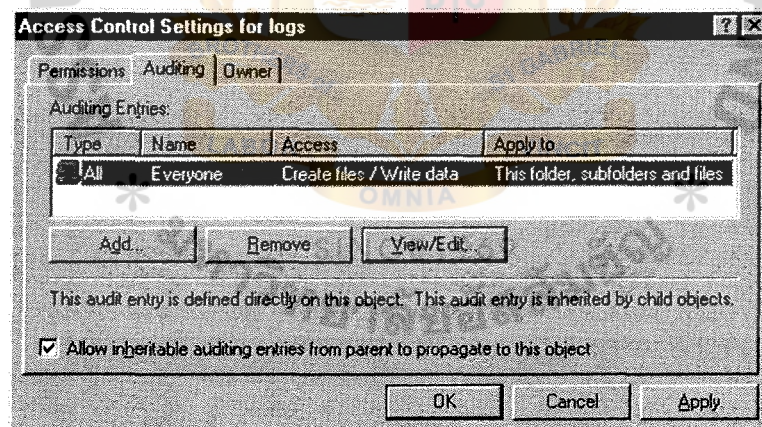


Figure 4.21. The ACL Editor in Windows 2000.

Auditing of the Registry

Registry auditing is configured using the 2000 Registry Editor (regedt32) application. Select the subkey you want to audit, then select Security → Auditing, as

shown in Figure 4.22. In Windows 2000, select Security → Permissions → Advanced → Auditing.

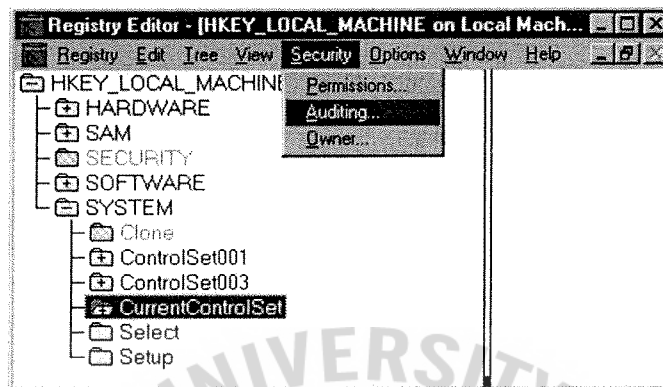


Figure 4.22. Configuring Auditing Using the Registry Editor.

I recommend that you enable auditing on the following registry keys:

HKEY_LOCAL_MACHINE\System

HKEY_LOCAL_MACHINE\Software

HKEY_CLASSES_ROOT

Enable auditing of successful and failed write attempts (Set Value, Create Subkey, Create Link, Delete, Write DAC) on these three keys, including all subkeys, as shown in Figure 4.23.

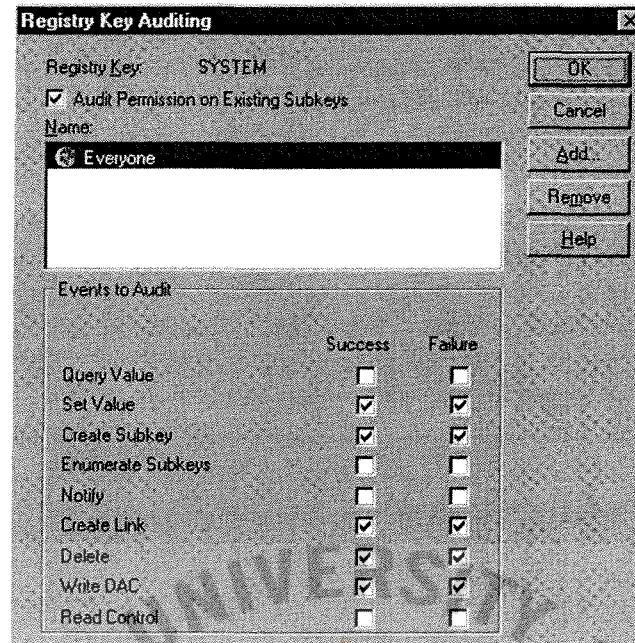


Figure 4.23. The Recommended Auditing Settings on HKLM\System.

4.5 Summary

Configuring Windows 2000 Server has several things to be concerned. By default setting might not be safe enough to protect the system from malicious attacks. Default installation leaves many ports and services opened, which hackers could use them in many ways to intrude the server. Normally, the system account database is not encrypted and this could be a threat for the system because the hacker might get hold of a system back up or an emergency repair disk then use a tool such as “L0phtCrack” to run a dictionary attack or a brute force attack on the System Account Manager (SAM) database to retrieve username and password, which the hacker could use it for his further tasks. In addition, the result from applying standard setting in account policy, no complex password requirement and no account threshold setting, the hacker could easily unveil the administrator’s username and password. User’s rights should be assigned relating to their specified tasks as the more rights are given, the more

possibilities the server would fail. TCP/IP filters configuration could also reduce the SYN flooding problem. Administrative tools and utilities as in normal installation could be used by hackers to attack the system, these tools and utilities should be moved and assigned permission that only administrators are allowed.

Compared to the old method using Telnet to administrate server which communicates by sending clear-text and was proved to be vulnerable, Windows 2000 terminal services is recommended for securing remote administration since it provides authentication and encryption.

There are three main backup's types as follows; full system backup, incremental backup, and differential backup. The appropriate type should be selected concentrating in hard disk space and time consuming. From time to time, the backup would have been restored on testing server to ensure that backup is working properly.

Auditing and monitoring systems are the important tools that help system administrator notice the irregular situations. From the last part of this chapter, the location of log file would better be stored in separated drive from the operating system.

V. CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

There are three main security issues which are intrusion, denial of services, information theft. The suggested solutions have been pointed out in order to increase securities and prevent Windows 2000 against those three main issues.

Windows 2000 has greatly added new features over Windows NT environments which both heighten the system securities and help users to work with ease.

System administrator needs to have appropriate plans for the server in the very beginning, including the policies to be used, authorization's level of control access, and operations to be implemented. System administrator may select to use whether by using pre-defined security templates or user-defined setting before exposing the server live on internet.

Periodical backup is a must, not just for security breaches which may arise at any circumstance, but also for unpredictable incidences in which causes the system no longer be able to operate. Backup should be made before and after new adjustment is applied. System should be monitored and log file should be reviewed in order to detect abnormal behavior promptly.

5.2 Recommendations

Keep in mind that the security could be breached through various methods. Though the solution of known issues is provided, however new security breaches are often detected as time goes by as well as serious bug issues, it is strongly recommended that system administrator should visit Microsoft's website to check for new updates regularly and subscribe the security mailing lists such as NTbugtraq (Discussion includes security exploit/bugs resolution on Windows NT, Windows 2000, third-party products, Macintosh, Netware, Unix related issues <http://www.ntbugtraq.com>) or

Windows 2000 Security Advice (Discussion of Windows 2000 security issues
<http://www.ntsecurity.net/>)



BIBLIOGRAPHY

1. Boswell, William. Inside Windows 2000 Server, 2nd Edition. Indiana: New Raiders, 2000.
2. Klevinsky, T. J., Scott Laliberate, and Ajay Gupta. Hack I.T. Security through Penetration Testing. Massachusetts: Pearson Education, Inc., 2002.
3. McClure, Stuart, Joel Scambray, and George Kurtz. Hacking Exposed Windows 2000: Network Security Secrets and Solutions, 3rd Edition. California: Osborne / McGraw-Hill, 2001.
4. Norberg, Stefan. Securing Windows NT/2000 Servers for the Internet. California: O'reilly & Associates, Inc., 2001.
5. Russell, Ryan, Bidwell Teri, Steudler Oliver, Walshaw Robin, and Huston L. Brent. Hack Proofing Your E-Commerce Site. Massachusetts: Syngress Publishing, Inc., 2001.
6. Scambray, Joel and Stuart McClur. Hacking Exposed Windows 2000: Network Security Secrets & Solution. California: Osborne/McGraw-Hill, 2001.
7. Spealman, Jill. MCSE Designing a Microsoft Windows 2000 Directory Services Infrastructure. Washington: Microsoft Press, 2001.
8. Todd, Chad and Johnson L. Norris. Hack Proofing Windows 2000 Server. Massachusetts: Syngress Publishing, Inc., 2001.

St. Gabriel's Library, Au

