Hardening Windows

Research Paper

An introduction to Network Security

Network Security is *never* absolute. If you're a CEO, don't let your IT-department tell you you're safe. If you're a network-administrator, don't think you are safe. If you're a user, don't trust the web.

Nevertheless, risks can be limited. Although there have been some security-problems with Windows NT, it still is one of the more secure operating systems. Personally, I would also use Unix for most of my tasks at hand, but this for entirely different reasons. When choosing for security, the choice is wide-open, unless you want some provisions like Mandatory-access control. When you demand such security measures, it suddenly becomes clear that the "normal" operating systems are almost of equal safety. I really make no distinction between a standard-Unix-install and a Windows-Install, except for the time it takes to harden them.

When you install an extra server on your network, the first thing to keep in mind, is to give this server on your network as much visibility as it really needs, and not more. Do not place internal DNS-servers on your DMZ when you do not really need them there, do not make all your internal boxes visible from the internet but place them behind Network Adress Translation or IP-Masquerading Routers, use a switched network topology as much as possible. Really, network-security is for a large part built out of a secure topology. It is necessary to understand where you will put a firewall, an IDS, an IPSEC or PKI-server.

When your topology has been well-considered, it is time to look at the individual hardening of the boxes. Make sure every box gets ALL of the necessary fixes installed at installation-time WITHOUT giving it internet-connectivity. It might be interesting to put a decently secured box on your network, give it access to the internet, get all the latest hotfixes, servicepacks, fixpacks and other updates, disconnect it physically from the internet, open up an FTP, and locally get all your fixes from that box. Next patch all boxes, harden them using the information you find in this document and many others (like the Microsoft Installation Guidelines and Checklists). Make sure to put an IPSEC-Portfilter on every machine, to make sure only legitimate connections are possible to your machine.

When you are here, you do not have a secure network. It is very interesting to understand that there are many 0-day-exploits on the loose for any operating system, any daemon, which are being applied in the wild by malicious users, right now! So, it is highly required for any systems administrator to keep an eye out for new security notifications, new hotfixes, new, better ways to make something more secure. Also, you should understand that host-security is not the only measure that needs to be supported from now on. Even though you might have a bright network-topology, it might still be vulnerable. New exploits for routing-protocols see the light of day frequently.

Also, much forgotten is the human factor in it all. Your network is secure... but when someone calls and starts yelling at an employee because he can't access this or that service and really needs it. This certain individual tells your employee that it might cost the employee's job if he fails to provide access to him. What does your employee care about most... the company-secrets... or his job ? Will he see through the mask of the "individual" behind the phone? Highly doubtful. Teaching your employees what should be done in such a case and that they really must be prepared for something like that to happen, is very important in todays society.

Last, but not least, comes the level of physical security. Some companies have high security in place on their systems, but put their servers in a publicly accessible room, most of them do it the other way around. Neither are good. To build a really secure system, you would have to have at least two people guarding it. One off-site, one on-site, so that local pressure on the individual's won't get an intruder very far, and off-site pressure alone also wouldn't be of any good. Never underestimate the human mind. Security-people can be cracked too.

As you can see, security is not something trivial. A really secure system is built in many layers... The network, the machine, the human. Personally, I do believe a network can be made impenetrable, but it would take up way too much resources to be really useful, and it would have to be of a very small size. I would really love having the challenge of once building one, however. Some people say complex security really isn't security, and every security measure should be visible and easy to monitor. I state this is not true. Security cannot be simple, because systems are not simple. Operating systems are some of the most complex pieces of software around. There aren't too many people who own a full understanding of the inside and outside of a certain system. For this reason, securing such a system cannot be simple either. You have to understand which security-component serves which goal. Simple security nowadays mostly represents putting a firewall, an IDS, and such highly visible components on your system, and monitor them. But as long as the operating systems don't have real, hard to break security-structures, this will not be effective. The future might have to be sought in other types of operating systems, which use capabilities instead of ACLs, which use Mandatory access control on top of Discriminatory Access Control, which support the highly underestimated "IP Security" bit of Ipv4 or run totally on Ipv6. The future of e-security is all up to us, but for the time being, we have to secure what once has been started, what has once been developed, by humans. And humans are by definition not simple.

Contents

Chapter 1 Hardening NT4

Steps to take during the installation

During the installation of NT 4, there are some things you should take into account which are quite important in securing your system. First of all, the filesystem. NT supports both FAT & NTFS. However, you can only enforce security-policies on directories and files on NTFS-filesystems. It is advised to put NT 4 as the only system on a production-server with ONLY NTFS-partitions.

Use a strong password when the administrator-password is requested during installation. It should be at least 9 characters long, and can be up to 14 characters. You should also use non-alphabetic characters in the first seven characters.

During installation, it is possible to create an extra user-account for routine-computer use. However, if you create a local account, it is automatically placed in the "Administrators"-group, so it has the ability to create other users. Keep this in mind.

Installing servicepacks & hotfixes

After installation of NT 4, make sure you have installed all drivers you are going to use (tapedrives, vga) and will not make any changes to this configuration soon. Now it is time to upgrade your NT installation to the highest servicepack level. Since Microsoft announced there will never be a SP 7 for NT, the latest fixpack is, and will remain to be SP6. You can find it at:
http://www.microsoft.com/ntserver/nts/downloads/recommended/sp6/allsp6.asp

After installation of the Service-pack, check Microsoft.com/security to see if there are any hotfixes released after SP6. There is also a Security Rollout Pack, SRP with all hotfixes since SP6a, which you might want to apply. It contains all fixes for:

Q241041 `Enabling NetBT to Open IP Ports Exclusively`

Q243649 `Unchecked Print Spooler Buffer May Expose System Vulnerability`

Q243835 `How to Prevent Predictable TCP/IP Initial Sequence Numbers`

Q244599 `Fixes Required in TCSEC C2 Security Evaluation Configuration for Windows NT 4.0 Service Pack 6a`

Q246045 `Malformed Resource Enumeration Arguments May Cause Named Pipes and Other System Services to Fail`

Q247869 `Local Procedure Call May Permit Unauthorized Account Usage`

Q248183 `Syskey Tool Reuses Keystream`

Q248185 `Security Identifier Enumeration Function in LSA May Not Handle Argument Properly`

Q248399 `Shared Workstation Setup May Permit Access to Recycle Bin Files`

Q249108 Registry Data Is Viewable By All Users During Rdisk Repair Update

Q249197 Internet Explorer Does Not Allow Use of Single SGC Certificate with 128-Bit Encryption for Virtual Sites

Q249863 SGC Connections May Fail from Domestic Clients

Q249973 Default RTF File Viewer Interrupts Normal Program Processing

Q250625 Default Registry Key Permissions May Allow Privilege Elevation

Q252463 Index Server Error Message Reveals Physical Location of Web Folders

Q257870 Malformed Print Request May Stop Windows 2000 TCP/IP Printing Service

Q259042 Handle Leak in WinLogon After Applying Windows NT 4.0 Service Pack 6

Q259496 Incorrect Registry Setting May Allow Cryptography Key Compromise

Q259622 Command Processor May Not Parse Excessive Arguments Properly

Q259728 Windows Hangs with Fragmented IP Datagrams

Q259773 Incorrect Response to Local Procedure Call Causes "Stop" Error Message

Q262388 Denial-of-Service Attack Possible from Linux RPC Client

Q262694 Malicious User Can Shut Down Computer Browser Service

Q264684 Patch for "Remote Registry Access Authentication" Vulnerability

Q265714 Windows NT 4.0 SNMP Registry Entries Are Readable

Q266433 Patch for Numerous Vulnerabilities in the LPC Port System Calls

Q267858 Memory Could Not Be Read Error Message While Doing File Operation

Q267861 RAS Registry Modification Allowed Without Administrative Rights

Q267864 MTS Package Administration Key Includes Information About Users

Q268082 DNS SOA Record May Reveal Administrator Account Name

Q269049 Registry-Invoked Programs Use Standard Search Path

Q269239 NetBIOS Vulnerability May Cause Duplicate Name on the Network Conflicts

Q271216 Fix for E-mail Issues Between 128-Bit and 56-Bit Encryption Using French Regional Settings

Q274835 Buffer Overflow in Network Monitor May Cause Vulnerability

Q275567 Multiple NetBT Sessions May Hang Local Host

Q276575 Patch Available for "Phone Book Service Buffer Overflow" Vulnerability

Q279336 Patch Available for Winsock Mutex Vulnerability

Q279843 Several Named Pipes Like NTSVCS and LSASS are Created Without Protection

Q280119 A Patch Is Available for the NTLMSSP Privilege Elevation Vulnerability

Q283001 Patch Available for Malformed PPTP Packet Stream Vulnerability

Q289246 Forged SID Could Result in Elevated Privileges in Windows NT 4.0

Q293818 Erroneous VeriSign-Issued Digital Certificates Pose Spoofing Hazard

Q294472 Index Server Search Function Contains Unchecked Buffer

Q296185 Patch Available for New Variant of the "Malformed Hit-Highlighting" Vulnerability

Q298012 Malformed RPC Request May Cause Service Failure

Q300972 Unchecked Buffer in Index Server ISAPI Extension Can Enable Web Server Compromise

Q303628 Relative Path Issue Can Run Program Under System Context

IIS Fixes Included in the SRP

Q188348 Specially-Malformed FTP Requests May Create Denial of Service

Q233335 Page Contents Visible When Certain Characters are at End of URL

Q234905 Improperly Formatted HTTP Request May Cause INETINFO Process to Fail

Q238349 Specially-Malformed Header in GET Request Creates Denial of Service

Q238606 Page Contents Visible When Certain Dot Extensions Present in the Virtual Directory Name

Q241805 Combined FTP and Domain Restriction Security Patch for IIS 4.0

Q244613 IIS 4.0 SSL ISAPI Filter Can Leak Single Buffer of Plaintext

Q246401 IIS May Improperly Parse Specific Escape Characters

Q249599 Virtual Directory Mapped to UNC Returns Server-Side Script Code When URL Contains Additional Characters at the End of the Request

Q252693 Chunked Encoding Request with No Data Causes IIS Memory Leak

Q254142 100% CPU Usage Occurs When You Send a Large Escape Sequence

Q260205 HTTP Request with a Large Number of Dots or Dot-Slashes Causes High CPU Utilization

Q260347 IIS 4: Fix for Cross-Site Scripting Issues

Q260838 IIS Stops Servicing HTR Requests

Q267559 GET on HTR File Can Cause a "Denial of Service" or Enable Directory Browsing

```
Q269862 Patch Released for Canonicalization Error Issue

Q271652 Patch Released for Malformed URL Vulnerability That Disables Web Server
Response

Q274149 Cookies Are Not Marked as Secure in IIS

Q277873 Patch Available for "Web Server File Request Parsing" Vulnerability

Q280322 FPSE: Patch for Malformed Web Form Submission Security Vulnerability

Q285985 Patch Available for New Variant of File Fragment Reading via .HTR
Vulnerability

Q295534 Superfluous Decoding Operation Can Allow Command Execution Through IIS
```

If you would like to have a C2-compliant system, you will also have to apply the C2 Update Hotfix, which can be found at microsoft.com.

Basic Security Measures

It is best to use the System key, and thus encrypt the password database by running syskey.exe (you can find it in the %systemroot%-directory). If you have the NT Resource Kit installed, you can run Passprop.exe to enforce use of strong passwords and to lockout the Administrator-account. When you are not using a serial modem/mouse or printer, remove the Serial and printer-ports from Control Panel/Ports /Devices

Remove the guest-account
Disable the Guest-account, and to make it harder for possible intruders to reactivate it, assign a very complex password to it and forbid logins from it 24h/24h.

Secure your shared printers
Printers receive EVERYONE-privileges by default, so any user which accesses your system by the internet, even without authentification, can make use of your printers. It is best to replace ALL "EVERYONE"-statements in your rights-assignment by "AUTHENTICATED USERS".

Password-protect your screensaver
The most overlooked part of Network-security is the fact that once an intruder gets inside physically, he can do almost everything. It is best to put a screensaver on every workstation on your network which starts after 3 minutes of inactivity, and put strong passwords on it. This can be defined in the Domain Policy so that it can't be changed on any system hooked up to your network.

File System Security

Since NT is a multi-user operating system, many different users may acquire access to your hard-drive in a authorized fashion. However, it is best not to have every user read the entire filesystem. Here are some basic-settings which you may use, however these need to be tuned for every system separately.

Directory	**Permissions**
\ (this is the root directory C:\)	**Administrators:** Full Control
	System: Full Control
	Authenticated Users: Read
\Boot.ini	**Administrators:** Full Control
\Ntdetect.com	**System:** Full Control
\Ntldr	**Authenticated Users:** Read
\Autoexec.bat	**Administrators:** Full Control
\Config.sys	**System:** Full Control
	Power Users: Change
	Authenticated Users: Read
\TEMP	**Administrators:** Full Control
	Creator Owner: Full Control
	System: Full Control
	Power Users: Change
	Authenticated Users: Special Directory Access-Read, Write, Execute, Special File Access: None
\WINNT and all subdirectories	**Administrators:** Full Control
	Creator Owner: Full Control
	Authenticated Users: Read, Execute
\WINNT\Repair	**Administrators:** Full Control
\WINNT\System32\config	**Administrators:** Full Control
	Creator Owner: Full Control
	System: Full Control

	Power Users: Change **Authenticated Users:** List
\WINNT\System32\spool	**Administrators:** Full Control **Creator Owner:** Full Control **System:** Full Control **Power Users:** Change **Authenticated Users:** Read
\WINNT\Cookies \WINNT\Forms \WINNT\History \WINNT\OCCache \WINNT\Profiles \WINNT\Sendto \WINNT\Temporary Internet Files \WINNT\Downloaded Program Files	**Administrators:** Full Control **Creator Owner:** Full Control **Authenticated Users:** Special Directory Access: -Read, Write, Execute, Special File Access: None **System:** Full Control

Disabling Services

It is best to remove every single service which you do not need. These are some you can remove safely, ofcourse depending on the applications you wish to deliver with your host:

- Alerter
- Clipbook server
- Computer Browser
- DHCP Client
- Directory Replicator
- Messenger
- Remote Procedure Call Locator
- Simple TCP/IP-Services
- SNMP Trap Service
- Spooler
- TCP/IP NetBIOS Helper
- Telephony Service

Remove Unnecessary Protocols

On an internet-server you probably only want the TCP/IP-protocol running. Running any other protocol can be a security risk. If you have, for example, IPX/SPX running, and there are boxes behind your network that run Novell, these might be accessible from the internet, depending on your routing-configuration. So, Disable NetBIOS Interface, Server and Workstation from the WINS-client, which can be found in your network-bindings, and make sure that the only protocol installed is TCP/IP.

Adjust the Security Policy

All security and authorization-requests on NT are handled by so-called security-policies. You decide what certain groups of users, or individual users can and may do. First of all, let's take care of the Account-policy.

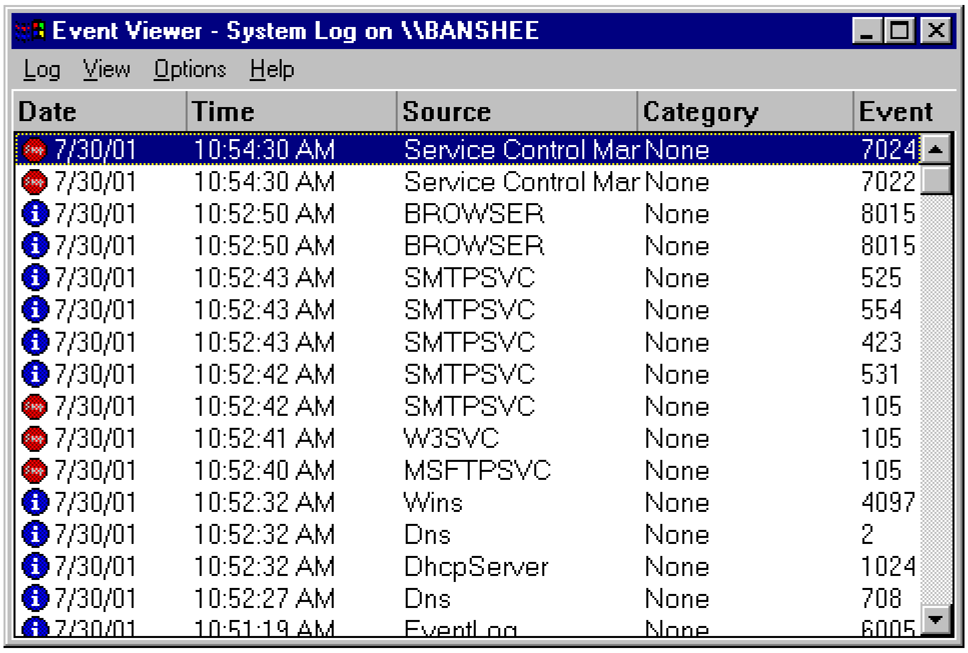
It is advised to rename the administrator-account, since on NT, due to a maximum password-length of 14 characters, brute-forcing isn't that hard.
You can also rename administrator, lock it out, and create a new account with full administrative privileges, since it might still be possible to enumerate the real administrator-account's name.

As password-policy, these are some important things:
- Passwords should expire every thirty days or less
- Minimum-password length should be 10 characters. Due to the way of hashing NT does, passwords smaller than 8 characters can be cracked more easily than long passwords.
- Accounts should be locked out after maximum 5 attempts (to block brute-forcing).
- The reset-count (time after which another logon-attempt will be counted in attempts) should be less than 30 minutes.
- Accounts should be locked forever if locked.
- Users have to login before they can change their passwords. Remote-changes should not be accepted.
- Password-changes have to be possible daily, or even more.
- Remember last 5 passwords

In the user-rights, these things apply:
- Only authenticated users may bypass transverse checking
- Remove all users their access to do a remote shutdown
- local logon is only possible for authenticated users and administrator
- shutdown may only be done by administrator

Auditing:
- Logon and Logoff, both success and failure
- File and Object Access, only failure
- User and group management, success and failure
- Security policy changes, success and failure
- Restart, shutdown, system failure

Audit-logs need to be checked daily... otherwise auditing really makes no sense at all.



Date	Time	Source	Category	Event
7/30/01	10:54:30 AM	Service Control Mar	None	7024
7/30/01	10:54:30 AM	Service Control Mar	None	7022
7/30/01	10:52:50 AM	BROWSER	None	8015
7/30/01	10:52:50 AM	BROWSER	None	8015
7/30/01	10:52:43 AM	SMTPSVC	None	525
7/30/01	10:52:43 AM	SMTPSVC	None	554
7/30/01	10:52:43 AM	SMTPSVC	None	423
7/30/01	10:52:42 AM	SMTPSVC	None	531
7/30/01	10:52:42 AM	SMTPSVC	None	105
7/30/01	10:52:41 AM	W3SVC	None	105
7/30/01	10:52:40 AM	MSFTPSVC	None	105
7/30/01	10:52:32 AM	Wins	None	4097
7/30/01	10:52:32 AM	Dns	None	2
7/30/01	10:52:32 AM	DhcpServer	None	1024
7/30/01	10:52:27 AM	Dns	None	708
7/30/01	10:51:19 AM	EventLog	None	6005

The Microsoft Windows NT Event Viewer

Make some changes in the registry

There are some basic registry-settings which should be changed after installing NT.

Displaying a legal notice
To avoid any problems later on, should your system get compromised, it is advised to make it very clear to possible attackers that they are doing something which will be punished, and to make it clear to the court that the attacker indeed was "entering" illegally, and not just stealing computertime, as most cases are trialled.
You should look in HKEY_LOCAL_MACHINE\Software\Microsoft\WindowsNT\CurrentVersion\Winlogon, at the key *LegalNoticeCaption*. Make sure this key contains a text which makes it clear that any trespasser is in violation, like:

Accessing this system without authorization is considered trespassing and will be prosecuted till the maximum extent possible under law. For more information contact myemail@mydomain.com

Hiding the name of last user logged on
By default, Windows NT will display the last user that has logged on in the User-textbox upon logon. This may come in handy on single-user systems, but in environments where many different users use these systems, it certainly has its security-related drawbacks. Bruteforcing a username just isn't necessary anymore. To disable it, go to *HKEY_LOCAL_MACHINE\Software\Microsoft\WindowsNT\CurrentVersion\Winlogon* in the registry, and put the value of the *DontDisplayLastUserName* key to 1.

Restricting anonymous access to the registry
You should change the *HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\LSA-key* to a dword with value 1. Then create a key *HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\SecurePipeServers\winreg*. You can leave it empty.

Enable SMB-signing
Samba-blocks should be signed. Change the following:

- *HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Rdr\Parameter*

Add the following keys:
EnableSecuritySignature, type REG_DWORD, Value 1
RequireSecuritySignature, type REG_DWORD, Value 1

Hide the machine in the Network Neighborhoord
When your machine isn't using netbios, you certainly don't want it to show up in Network Neighborhood-lists. Disable it in the registry using the HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters\Hidden-key. This should be a DWORD with value 1.

Disable Default Admin Shares
Even when you do not create any shares, NT always shares some standard-things, called the administrator-shares. It's best to disable them with the following keys:

- *HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters\EnableSharedNetDrives*, type REG_DWORD, value 0

- *HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters\AutoAdminWKS*, type REG_DWORD, value 0

Disable LanMan Password Hash support

Change HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\LMCompatibilityLevel to value 2 (dword).

Erase Pagefile on Clean Shutdown

Change HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management\ClearPageFileAtShutdown to value 1 (dword).

Allocate Floppies and CD-ROM

- Change HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\Winlogon\AllocateFloppies to value 1 (reg_sz).
- Change HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\Winlogon\AllocateCDRoms to value 1 (reg_sz).

Disable AutoRun on CD

Change HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Cdrom\autorun to value 0 (dword).

Enable Full Privilege Auditing

Change HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\FullPrivilegeAuditing to value 1 (reg_binary).

Restrict Event Log Access

- Change HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EventLog\Application\RestrictGuestAccess to value 1 (dword).
- Change HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EventLog\Security\RestrictGuestAccess to value 1 (dword).
- Change HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EventLog\System\RestrictGuestAccess to value 1 (dword).

Removing the Posix and OS/2-subsystems

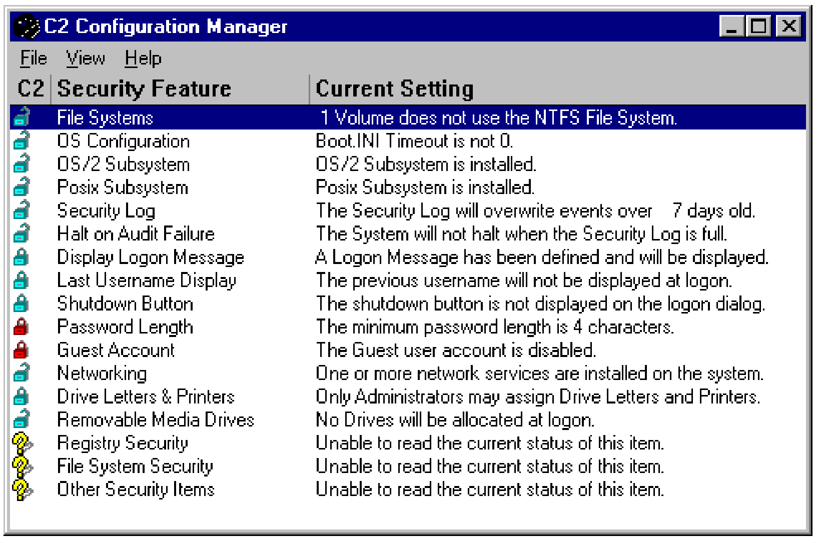
The roots of Windows NT can be found back in the early 90's when Microsoft was still working together with IBM on the OS/2 Warp Operating System. For this reason, there are still OS/2 and Posix (unix) subsystems in Windows 2000. No security-related problems have been found in these subsystems yet, but it is best to remove them to be certain. This can be done by removing the OS/2 and Posix registry-values from the following key:

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\SubSystems.

Afterwards, remove the os2*, posix* and psx* files in %systemroot%\System32

Tune C2 Level Configuration

NT has been marketed by Microsoft as a C2-certified system. No NT-installation however is certified. When correctly secured, they are compliant, but need certification before they really are certified ;) However, it is never bad to try and reach this level of security, so, if you want to do a decent checkup of your system, get the NT Resource Kit. Included is a program, called the "C2 Level Configuration Tool". It will do a complete check and notify you of things that can be changed to make it more secure.



C2	Security Feature	Current Setting
	File Systems	1 Volume does not use the NTFS File System.
	OS Configuration	Boot.INI Timeout is not 0.
	OS/2 Subsystem	OS/2 Subsystem is installed.
	Posix Subsystem	Posix Subsystem is installed.
	Security Log	The Security Log will overwrite events over 7 days old.
	Halt on Audit Failure	The System will not halt when the Security Log is full.
	Display Logon Message	A Logon Message has been defined and will be displayed.
	Last Username Display	The previous username will not be displayed at logon.
	Shutdown Button	The shutdown button is not displayed on the logon dialog.
	Password Length	The minimum password length is 4 characters.
	Guest Account	The Guest user account is disabled.
	Networking	One or more network services are installed on the system.
	Drive Letters & Printers	Only Administrators may assign Drive Letters and Printers.
	Removable Media Drives	No Drives will be allocated at logon.
	Registry Security	Unable to read the current status of this item.
	File System Security	Unable to read the current status of this item.
	Other Security Items	Unable to read the current status of this item.

Renaming System Tools

It is wise to rename all system-tools, like the command-interpreter, to arbitrary names. Why's that, you probably wonder... isn't that a bit like security by obscurity?

Partially, you're right. And security by obscurity should never be a goal as such, but can be helpful to keep your things safe.
That however, isn't the only reason why we would rename system-commands. Most attackers are just plain script-kiddies. They use pre-written exploits to take advantage of your system. Those exploits are, for the most part, programs which execute buffer-overflows on your system, and execute the necessary code to spawn a shell on a port (shellcode). However, since a buffer-overflow can only contain a few bytes of code to execute, shellcode can't be enormously large. Most hackers trust that your command-interpreter is called command.com, cmd.exe or cmd32.exe, and is based in c:\winnt\system32. If your interpreter is located somewhere else, 99% of all exploits we know today won't work.

To rename them, make a directory, for example d:\toolkit, put access-control on this directory so only administrator may use the executables in this directory. Then you move the most important files, like fdisk, rdisk, command.com, cmd.exe to this directory. Also remove the COMSPEC-variable, or leave it pointed to c:\winnt\system32, so exploits which check the environment will be misleaded. As you probably guessed, your flashy "MS-DOS"-icon will not work anymore. However, I advise you not to move the link to the file on this icon, because otherwise users can (possibly) read where it points to, and obtain the right command-interpreter this way. However, since they're not administrator yet, they won't be able to execute it (thank you ACL).

Chapter 2 Hardening W2K

Steps to take during the installation

During the installation of Windows 2000, there are some things you should take into account which are quite important in securing your system. First of all, the filesystem. Win2k supports both FAT16, FAT32 & NTFS. However, you can only run security-policy's on directories and files on NTFS-filesystems. It is advised to put Win2k as the only system on a production-server with ONLY NTFS-partitions.

Use a strong password when the administrator-password is requested during installation. It should be at least 9 characters long, and can be up to 120 characters. You should also use non-alphabetic characters in the first seven characters. If you do not have any NT4 / Win95 / Win98-clients, only use passwords longer than 14 characters. In this case, no (insecure) LAN Manager-strings will be built. However, as said before, only Win2k-clients will be able to connect to your server.

Installing Servicepacks & hotfixes

After installation of Win2k, make sure you have installed all drivers you are going to use (tapedrives, vga) and will not make any changes to this configuration soon. When this is done, you are ready to upgrade to the most recent servicepack. At time of writing, the most recent service-pack for Windows 2000 is SP2. You can find it @ Microsoft.com. If you have a system with SP1 applied, you do not need to remove it. If your system is a clean Windows 2000 install, you can install SP2 straightaway, you do not need to install SP1 first.

After installation, check Microsoft Security to see if there are any Hotfixes which need to be applied.

Basic Security Measures

First of all, during installation, it is very important to have a clear view at what you will be doing with the system. If you're planning to run a very public webserver, prepare different partitions for the webroot, for logfiles, and the system. You can easily place the system, including your webserver on for example the C:-partition, but in that case, move your webroot later on to another partition. The reason for putting logfiles on a separate partition is to prevent denial of service-attacks by flooding the logfiles.

Second, make sure you know all your users. In this case, you will notice when strange users get added to the system. This is an old unix-value, but it still guarantees at least a "sense" of security around your system, which even your users will feel. You know your system, you know who uses it, you will know who and when someone breaches it.

Read your logfiles! It is advised to log as much as you can use. This means, that you shouldn't be flooding your log yourselves by auditing all kind of useless stuff, like the usage of every, every object on your system, even if it is authorized. When logs grow too large, system-administrators usually get quite careless in looking over them. You will have to find out for yourself how much information you can process every day. Log-files should be checked daily, and if possibly, automatically archived to another location, preferably on another, hardened system, which isn't reachable from any untrusted network. You can even log to Unix-servers with different kinds of syslogd-ports... Encrypted logging over the network is another option. Just in case your intruder is sniffing.

File System Security

Since Win2k is a multi-user operating system, many different users may acquire access to your hard-drive. However, it is best not to make it possible for every user to read the entire filesystem. Here are some basic-settings which you may use, however these need to be tuned for every system separately.

Directory	**Permissions**
\ (this is the root directory C:\)	**Administrators:** Full Control **System:** Full Control **Authenticated Users:** Read
\Boot.ini \Ntdetect.com \Ntldr	**Administrators:** Full Control **System:** Full Control **Authenticated Users:** Read
\Autoexec.bat \Config.sys	**Administrators:** Full Control **System:** Full Control **Power Users:** Change **Authenticated Users:** Read
\TEMP	**Administrators:** Full Control **Creator Owner:** Full Control **System:** Full Control **Power Users:** Change **Authenticated Users:** Special Directory Access-Read, Write, Execute, Special File Access: None
\WINNT and all subdirectories	**Administrators:** Full Control **Creator Owner:** Full Control **Authenticated Users:** Read, Execute
\WINNT\Repair	**Administrators:** Full Control
\WINNT\System32\config	**Administrators:** Full Control **Creator Owner:** Full Control **System:** Full Control **Power Users:** Change **Authenticated Users:** List

\WINNT\System32\spool	**Administrators:** Full Control **Creator Owner:** Full Control **System:** Full Control **Power Users:** Change **Authenticated Users:** Read
\WINNT\Cookies \WINNT\Forms \WINNT\History \WINNT\OCCache \WINNT\Profiles \WINNT\Sendto \WINNT\Temporary Internet Files \WINNT\Downloaded Program Files	**Administrators:** Full Control **Creator Owner:** Full Control **Authenticated Users:** Special Directory Access: -Read, Write, Execute, Special File Access: None **System:** Full Control

Disabling Services

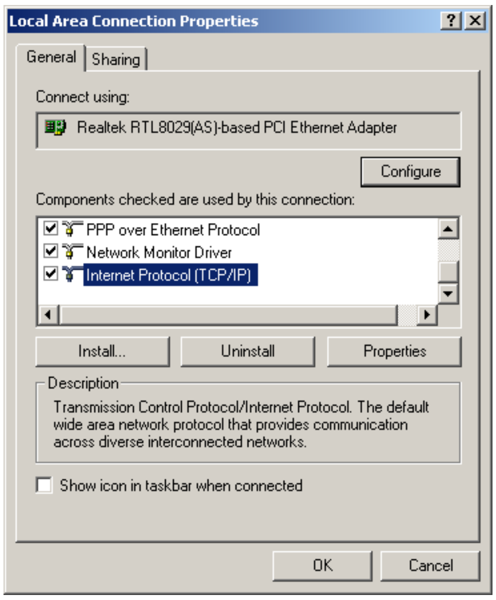
After installing Windows 2000, it is best to disable all services which are not necessary for the system to run. After this is done, you can gradually add services which are needed. This is the absolute minimum framework for a windows 2000-system to run:

- Computer Browser
- NTLM SSP
- DNS Client
- Eventlog
- Logical Disk Manager
- Plug & Play
- Protected Storage
- Security Accounts Manager

Now start adding services till you acquire correct operation of the system. If you want to run a webserver for example, activate the "Web Publishing Service". Some services require that other services are started too, these are called "dependencies", so it might be a bit of a nasty work to get the system running with the minimum of services, and still reach the desired level of functionality.

Disabling a service too much might have all sorts of strange consequences for the system. For example, you might suddenly experience icons appearing in windows were normally the accompanying text should appear... for example in the control-panel, so you are not able to reactivate the service. If you ever find yourself to be in this case, run regedit.exe and go to
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\
Here you should try to change the start-value of the services which you had disabled earlier to either 2 or 3 and reboot your system. This should help in most cases.

Remove Unnecessary Protocols


Local Area Connection Properties

General | Sharing

Connect using:

Realtek RTL8029(AS)-based PCI Ethernet Adapter

Configure

Components checked are used by this connection:

- [✓] PPP over Ethernet Protocol
- [✓] Network Monitor Driver
- [✓] Internet Protocol (TCP/IP)

Install... | Uninstall | Properties

Description

Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

[ ] Show icon in taskbar when connected

OK | Cancel

On an internet-server you probably only want the TCP/IP-protocol running. Running any other protocols can be a security risk. If you have, for example, IPX/SPX running, and there are boxes behind your network that run Novell, these might be accessible from the internet, depending on your routing. So, Disable NetBIOS Interface, Server and Workstation from the WINS-client, which can be found in your network-bindings, and make sure that the only protocol installed and bound to your network-interfaces is TCP/IP.

Remove Default Shares

Windows 2000 has a lot of Default Shares which cannot easily be disabled. Disabling them may cause some programs to fail, so it is advised not to do this in a production-environment without prior testing. These shares include IPC$, print$, c$, d$ (for the root of each fat/fat32/ntfs-partition), Admin$, Fax$ and Netlogon.

By disabling the Server-service as we did previously, these are automatically inactive. It is however better to disable them completely, as to prevent usage of them when a malicious user starts the server-service anyway.

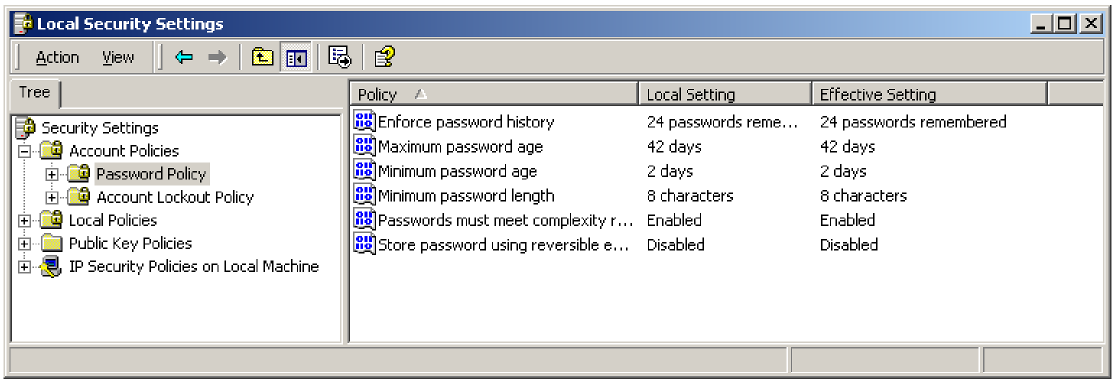
You can disable them in this registry-key:

```
HKeyLocal Machine\SYSTEM\CurrentControlSet\Services\LanManServer\Parameters
```

Adjust the Security Policy

All security and authorization-requests on Win2k are handled by so-called security-policies. You decide what certain groups of users, or individual users can and may do. First of all, let's take care of the Account-policy.

It is advised to rename the administrator-account, brute-forcing isn't impossible. You can also rename administrator, lock it out, and create a new account with full administrative privileges, since it might still be possible to enumerate the real administrator-account's name.


Local Security Settings

Policy	Local Setting	Effective Setting
Enforce password history	24 passwords reme...	24 passwords remembered
Maximum password age	42 days	42 days
Minimum password age	2 days	2 days
Minimum password length	8 characters	8 characters
Passwords must meet complexity r...	Enabled	Enabled
Store password using reversible e...	Disabled	Disabled

Some guidelines for a correct password-policy:
- Passwords should expire every thirty days or less
- Minimum-password length should be 10 characters. Due to the way of hashing Win2k does, passwords smaller than 8 characters can be cracked more easily than long passwords. If you only have clients running windows-2000, it is advised to make passwords at least 14 characters long, this prevents the building of Lan Manager-password-hashes.
- Accounts should be locked out after maximum 5 attempts (to block brute-forcing).
- The reset-count (time after which another logon-attempt will be counted in attempts) should be less than 30 minutes.
- Accounts should be locked forever if locked.
- Users have to login before they can change their passwords. Remote-changes should not be accepted.
- Password-changes have to be possible daily, or even more.
- Remember last 5 passwords

Concerning user-rights, these things apply:
- Only authenticated users may bypass transverse checking
- Remove all users their access to do a remote shutdown
- local logon is only possible for authenticated users and administrator
- shutdown may only be done by administrator

Security Options
- Restrict CD-ROM access to the locally logged on user only : Enabled
- Restrict floppy access to the locally logged on user only : Enabled
  In the summer of 2001, some problems were found concerning local disk-access on the windows-platform. Due to many calls to the drive, it was possible to cause a DoS of multiple web & ftp-servers. Setting these options prevents such attacks.
- Clear virtual memory pagefile when system shuts down : Enabled
  The Windows Operating System uses a swapfile to manage memory. When something isn't used for that long, it is transferred from RAM to the swapfile, and exchanged for more frequently-used information. This has a great influence on the performance of the server. Some programs however store sensitive information in the RAM, and this might get moved to hard-disk. For this reason, it is advised to clear the swapfile before shutdown.
- Additional Restrictions for anonymous connections :
  No access without explicit anonymous permissions
- Allow system to be shut down without having to log on: Disabled
- Audit use of backup and restore privilege: Enabled
- Do not display last user name in logon screen: Enabled
- Message Text for Users Attempting to Log on:
  All unauthorized access to this system will be prosecuted till maximum extent possible under law.
- Message Title for users attempting to log on: A T T E N T I O N !

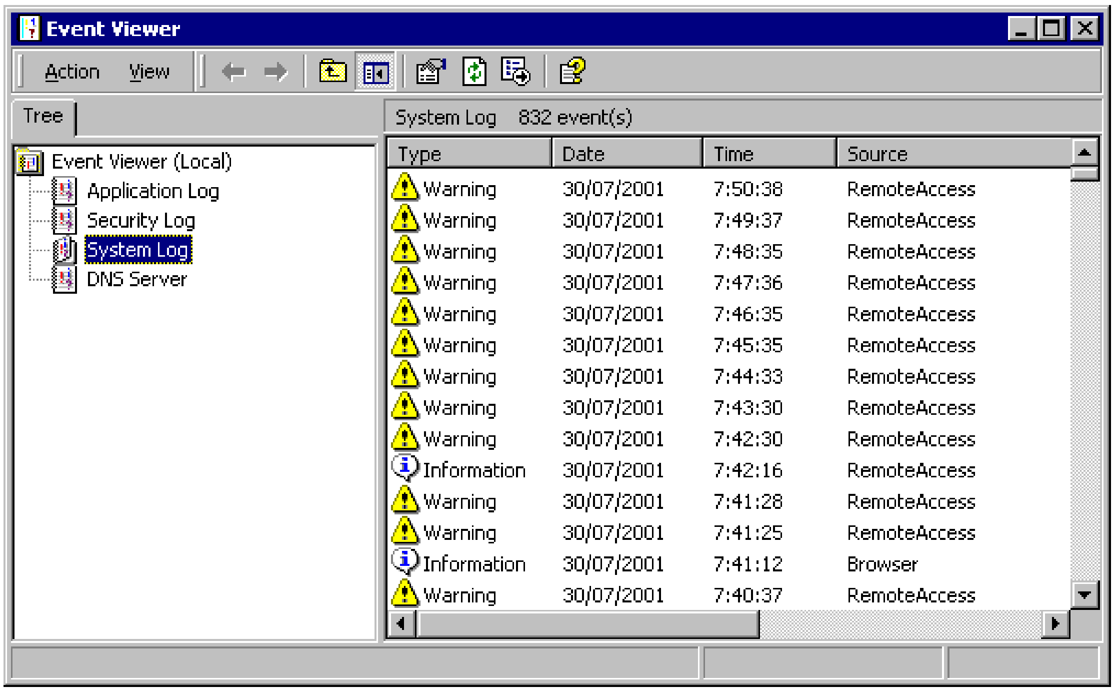
- Number of previous logons to cache: 0
- Prevent users from installing printer drivers: Enabled
- Recovery Console: Allow automatic administrative logon: Disabled
- Rename Administrator-account: Fill in an inexistant user
  After you have done this, the administrator-account will be changed into the is name. This prevents brute-forcing-attacks against username "administrator". Please note that if null-sessions to your server are allowed, it is still possible to enumerate the SID of administrator.
  You might even think about making a new account "administrator", with NO rights at all, to monitor usage and attack patterns against this account. Fully audit every attempt to logon using this, fake, account.
- Send unencrypted password to connect to third-party SMB Servers: Disabled
  In some cases it might be necessary to allow this, for example when you run client-operating-systems who do not support encrypted SMB-passwords.
- Shutdown system immediately if unable to log security audits: Enabled
  Make sure that the partition containing your logs is big enough, to prevent denial-of-service-attacks by flooding the logfiles.
- Strengthen Default Permissions of Global system objects: enabled
- Unsigned Driver Installation Behavior: do not allow
- Unsigned Non-driver Installation Behavior: do not allow

Some things that should be changed if you are using only Win2k-clients
These are some basic settings that can be set a bit safer, if you are only using Win2k-clients. You will not be able to use any Win9x-, Linux-, BSD- or OS/2-clients.

- Secure Channel: Require Strong (Windows 2000 or Later) Session Key
- Digitally Sign Client Communication (Always): Enabled
- Digitally Sign Server Communication (Always): Enabled

Auditing:
- Logon and Logoff, both success and failure
- File and Object Access, only failure
- User and group management, success and failure
- Security policy changes, success and failure
- Restart, shutdown, system failure

Audit-logs need to be checked daily... otherwise auditing really makes no sense at all.



System Log 832 event(s)

Type	Date	Time	Source
Warning	30/07/2001	7:50:38	RemoteAccess
Warning	30/07/2001	7:49:37	RemoteAccess
Warning	30/07/2001	7:48:35	RemoteAccess
Warning	30/07/2001	7:47:36	RemoteAccess
Warning	30/07/2001	7:46:35	RemoteAccess
Warning	30/07/2001	7:45:35	RemoteAccess
Warning	30/07/2001	7:44:33	RemoteAccess
Warning	30/07/2001	7:43:30	RemoteAccess
Warning	30/07/2001	7:42:30	RemoteAccess
Information	30/07/2001	7:42:16	RemoteAccess
Warning	30/07/2001	7:41:28	RemoteAccess
Warning	30/07/2001	7:41:25	RemoteAccess
Information	30/07/2001	7:41:12	Browser
Warning	30/07/2001	7:40:37	RemoteAccess

The Windows 2000 Event Viewer

Make some changes in the registry

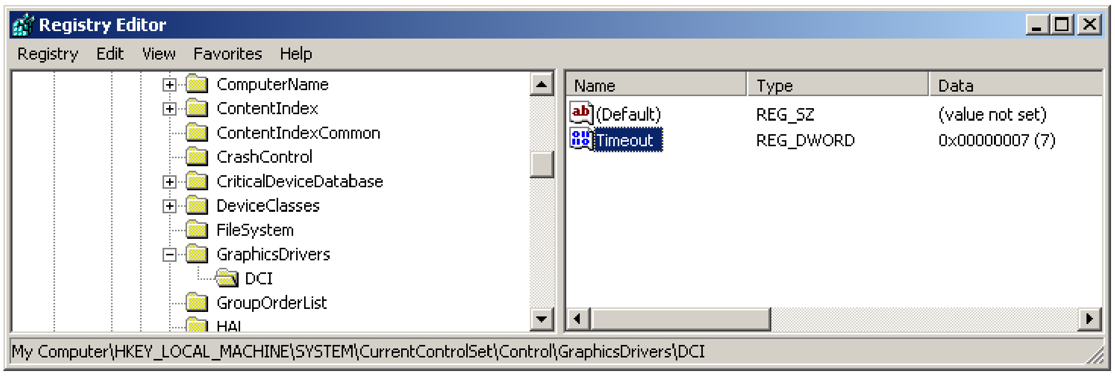
Removing OS/2 and Posix-subsystems

The roots of Windows 2000 can be found back in the early 90's when Microsoft was still working together with IBM on the OS/2 Warp Operating System. For this reason, there are still OS/2 and Posix (unix) subsystems in Windows 2000. No security-related problems have been found in these subsystems yet, but it is best to remove them to be certain. This can be done by removing the OS/2 and Posix registry-values from the following key:
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\SubSystems .
Afterwards, remove the os2*, posix* and psx* files in %systemroot%\System32

Some problems have been reported in removing them, so be careful, and take a copy of your registry beforehand. If you only remove the files it will also be disabled, however, an intruder might put them back in their place.

Disable DirectDraw

DirectDraw is in direct violation with almost everything C2 stands for. If you really require a secure system, you should disable it. This will make it next to impossible to run any games on the server, but I hardly think this is is a negative point.
You may disable it by changing the "Timeout"-key in
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\GraphicsDrivers\DCI and to 0.



Name	Type	Data
(Default)	REG_SZ	(value not set)
Timeout	REG_DWORD	0x00000007 (7)

My Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\GraphicsDrivers\DCI

Hardening the TCP/IP-stack

Windows 2000 has some very fine options which can make the TCP/IP-stack a lot harder to break, protect against Denial-of-Service attacks, and make the stack function as a low-level host-based firewall. Really interesting things if you want a secure system.

First of all, we're gonna toughen up the stack a bit. On 2k, this is done in the registry, all under HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpip\Parameters

- **SynAttackProtect**: this is the time the system will wait for SYN-ACKs, protecting itself from a SYN-flood. After a clean-install it should be set to 0, which means no protection whatsoever. 1 reduces retransmission retries and delays the route cache entry a bit. 2 does the same as 1, but also gives a bit of delay at winsock-level. We advise you set this REG_DWORD to 2.
- **TcpMaxHalfOpen**: determines the number of connections in SYN_RECEIVED state allowed before protection against SYN-attacks starts to operate. It is a DWORD with a value from 100-0xFFFF. On Win2k Pro, the default is 100, and 500 for Win2k Advanced server.
- **TcpMaxHalfOpenRetried**: determines the number of SYN_RECEIVED state allowed for which there has been at least one retransmission of the SYN sent before syn-attack protection begins to operate. DWORD with a value between 80-0xFFFF. The default for Win2k Pro is 80, leading up to 400 for Win2k Advanced Server.
- **PerformRouterDiscovery**: Should Windows 2000 try to perform router-discovery? When this is 0, it won't, on 1 router-discovery is enabled and with value 2 the setting is controlled through received DHCP-info. We advise you to use 0 (to prevent the system from trusting unknown routers).
- **EnableICMPRedirects**: Should Windows 2000 alter its routing table when an ICMP redirect is received ? Default this is set to 1 which means "yes", but the recommended value is 0, so spoofed ICMP cannot alter the routing table.
- **KeepAliveTime**:How often should the TCP-stack attempt to verify if an idle connection is alive by sending a keep-alive-packet? Value between 1-0xFFFFFFFF milliseconds. Normally this runs up to 7 200 000, which is two hours, but a much shorter value is advised, like 200 000.

Advanced packet-filtering using Win2k
Win2k is quite exceptional since it contains two built-in packetfilters. These can be used to block traffic to your machine on ports where there shouldn't be any traffic.

One packet-filter is only very rudimentary, a little bit less safe, and harder to configure. The other one, IPSec Port Filtering, is probably one of the most powerful means in protecting your windows-server. Above that, it can even be administered from the command-line (if you have the Win2k Server Resource Kit), which makes it an ideal choice for a web/application-server.
TCP/IP-filtering
With TCP/IP-filtering you can block access to all ports, and then choose a few ports which you would like to allow inbound access to. TCP/IP-filtering does not take into account any sessions initiated from the inside, so these are never blocked. You can find TCP/IP-filtering on every dial-up-connection, so take a look in "Network and dial-

up connections", select the interface to apply the filter on, then choose "properties", "general", "internet protocol(TCP/IP)", "properties", "advanced", "options", "TCP/IP-filtering". Now you select "enable TCP/IP-filtering" and make a selection of ports you wish to allow.
Note: you cannot block ICMP through this interface.

IPSec Port Filtering
The fun part. IPSec port-filtering is actually quite an advanced firewall. There are two ways to configure it, *Command-line-based* and from within a *graphical user interface*. If you have any experience with Linux IPCHAINS / IPTABLES, this packetfilter is quite comparable with that implementation. You have to create **filter lists** & **filter actions**.
First of all, you will create a policy (a chain in linux) which contains the ipsec filtering-actions that you want applied on all network-traffic to / from this system.

- *Graphical User Interface*

You can find the GUI for IPSec Port Filtering in Administrative Tools / Security Policy (group or local) / IP Security Policies on Local Machine. You give a policy a name, and then add filters you want to be part of that policy. Afterwards you just choose "OK" and your policy will be applied.

- *Textmode Interface*

To use the textmode-interface you will be needing ipsecpol.exe, a file included with the NT Resource Kit. To demonstrate this tool, I will show some keystrokes which will create a security-policy.

```
ipsecpol -w REG -p "Webserver" -r "Inbound web" -f *+216.123.24.25:80:TCP -f
*+216.123.24.25:443:TCP -n PASS
ipsecpol -w REG -p "Webserver" -r "Block other inbound" -f *+216.123.24.25 -n BLOCK
ipsecpol -w REG -p "Webserver" -x
```

This ought to get you started. "-f" means the definition fo the filter, "-p" is the policy-name, "-r" defines the name of the ruleset, "-n" tells the system what to do, PASS or BLOCK the packet. The * means that the rule is mirrored.

You can also remotely activate a policy on a machine by using
```
ipsecpol \\servername -w REG -p "name of policy" –x
```

Renaming System-tools

It is wise to rename all system-tools, like the command-interpretor, to arbitrary names. Why's that, you probably wonder... isn't that a bit like security by obscurity?

Partially, you're right. And security by obscurity should never be a goal as such, but can be helpful to keep your things safe. That however, isn't the only reason why we would rename system-commands. Most attackers are just plain script-kiddies. They use pre-written exploits to take advantage of your system. Those exploits are, for the most part, programs which execute buffer-overflows on your system, and execute the necessary code to spawn a shell on a port (shellcode). However, since a buffer-overflow can only contain a few bytes of code to execute, shellcode can't be enormously large. Most hackers trust that your command-interpreter is called command.com, cmd.exe or cmd32.exe and is based in c:\winnt\system32. If your interpreter is located somewhere else, 99% of all known exploits up till today will not work.

The main problem with Win2k and securing those binaries is a built-in-security measure, called "Windows System File Protection". In itself this was a good step from microsoft, since it was meant to protect important system-files against trojans, badly written software and software that tries to replace system-files with older versions of those files during installation. However, Windows System File Protection will also stop you from moving the binaries. Automatically, when you move them, they will be replaced by Win2k itself. Now, we do not guarantee this will work, cause on a few 2k-boxes here Windows System File Protection remained active, but here is the most-used way to remove Windows System File Protection [WSFP].

According to Microsoft, WSFP should be turned off when you create a SFCDisable-value in the HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon-key. This should contain value 1 to disable it, but Microsoft notes that this option only works when you are using a serial-debugger. I'm afraid most of us aren't. However, diverse other sources report the fact that if you put a REG_DWORD value of "ffffff9d" in this key, it is turned off permanently. This worked for me on most boxes, but not all. The ignorant ones were wisely upgraded to OpenBSD.
However, there is one last approach if the above method fails. Since the backups of the systemfiles are kept in %systemroot%\system32\Dllcache, you can delete them before replacing the file. Now SFP won't find the backup, and will not be able to replace the changed files with their backup.

To rename them, make a directory, for example d:\toolkit, put access-control on this directory so only administrator may use the executables in this directory. Then you move the most important files, like fdisk, rdisk, command.com, cmd.exe to this directory. Also remove the COMSPEC-variable, or leave it pointed to c:\winnt\system32, so exploits will be misleaded. As you probably guessed, your flashy "MS-DOS"-icon will not work anymore. However, I advise you not to move the link to the file on this icon, because otherwise users can (possibly) read where it points to, and obtain the right command-interpreter this way. However, since they're not administrator yet, they won't be able to execute it (thank you ACL).

As a last step here, it might be interesting to audit ALL use of these executables. Since these (if not used in any scripts) are fairy little used commands on a production-server, auditing them will give you a notion right-away when someone is abusing your system.

Microsoft EFS : Encrypted File System

Microsoft EFS probably is one of the least used security-measures integrated in Windows 2000. Most administrators forget that when a malicious user can get hold of your hard-drive physically, all ACL-lists are useless. They insert the drive into a box, boot Linux, BSD or some other operating-system with support for NTFS/2k, and read all your files without any problems.

To prevent sensitive information from falling into the wrong hands, EFS was integrated into Windows 2000. EFS, or the Encrypted File System, delivers a form of Encryption to the desktop. You can decide which drive, file or directory is encrypted by right-clicking on it, going to the Properties-section and click on the "Advanced"-Button. Here you can choose to "*Encrypt Contents to Secure Data*". All encryption/decryption occurs transparently when you access the file, so you will not be asked for passwords, this is all done using your normal Windows 2000-password.

When integrating EFS in a Windows 2000 Networking-environment, it is important to remember that when you open a decrypted file on another host on the network, this file will be decrypted on the other machine, and all data will pass the wire in cleartext.

Microsoft published some "Best Practices Guidelines", which say:

- Every user's "my documents"-folder should be encrypted
- Teach users NOT to encrypt individual files but only folders. Since programs work on files in different, mysterious ways, no files will be decrypted unexpectedly.
- Print Spool Files should be generated in an encrypted directory. The problem is that under Win2k, system-directories, like the spooler-directory, this directory should be moved to another location which can be encrypted.
- Recovery "Best Practices":
    - Private keys assigned to recovery certificates are extremely sensitive. They should be generated on a physically secured machine or exported to a PFX-file under heavy encryption, stored on a secured disk.
    - Recovery agent certificates should be assigned to special recovery agent accounts that are not used for any other purpose.
    - Do not destroy recovery certificates or private keys when recovery agents are changed (which should occur periodically). Keep all of them, until all files that may have been encrypted with them are updated.
    - Designate two or more recovery agent accounts per Organizational Unit (OU), depending on the size of the OU. Designate two or more computer for recovery, one for each designated recovery agent account, and give permissions to appropriate administrators to use the recovery agent accounts.
    - Implement a recovery agent archive program to ensure that encrypted files can be recovered using obsolete recover keys. Recovery

certificates and private keys must be exported and stored in a controlled and secure manner. Ideally, as with all secure data, archives should be stored in a controlled access vault and you should have two archives: a master and a backup. The master is kept on-site, while the backup is located in a secure off-site location.

Chapter 3 Hardening IIS

IIS is probably the most used web-server for the Microsoft Windows-platform, and with good reason. Although there have been some security-problems in the past with IIS, no other webserver has been able to beat it concerning plain web-serving speeds, on Win32 that is. IIS however is more than just a webserver. It also contains facilities for SMTP & NNTP, better known as the mail-transfer-protocol and the news-transfer protocol. At this time, hese issues will not be adressed in the guide, mainly due to the fact that these are not as widely deployed as the IIS Webserver. If you are not planning on using them, DISABLE the SMTP & NNTP services!

Following the guidelines mentioned hereunder, IIS can be made immune for most exploits that are currently in existance, and against a whole set of bugs which will probably be unleashed in the future. A secure server is secure by topology, not really by the type and number of bugs it contains.

A webserver should only have access to its own content, not to other software/data on the system. That knowledge will be applied here to bring your webserver up to a secure level. This concept is called "*least privilege*"

Setting appropriate ACL's on virtual directories

It is of the utmost importance to decide which user is able to run / view / execute certain types of files. If a malicious user is capable of reading the contents of one of your ASP-scripts, he has a far bigger chance to detect errors in your code, which might even be used in further penetrations of your site. Also, this differs a lot from application to application... so these are just some basic guidelines.

File Type	**Access Control Lists**
CGI (.exe, .dll, .cmd, .pl)	Everyone (X) Administrators (Full Control) System (Full Control)
Script files (.asp)	Everyone (X) Administrators (Full Control) System (Full Control)
Include files (.inc, .shtm, .shtml)	Everyone (X) Administrators (Full Control) System (Full Control)
Static content (.txt, .gif, .jpg, .html)	Everyone (R) Administrators (Full Control) System (Full Control)

Microsoft advises, instead of putting ACL-privileges on each file seperately, to put all files with a specified file-type in a different directory. Then you set the ACL on that directory, and allow files to inherit the contents of the ACL of their mother-directory. You might, for example, want to create:

- c:\inetpub\wwwroot\static (.html)

- c:\inetpub\wwwroot\include (.inc)
- c:\inetpub\wwwroot\script (.asp)
- c:\inetpub\wwwroot\executable (.dll)
- c:\inetpub\wwwroot\images (.gif, .jpg)

Two directories will need our special attention, those for the FTP-server, and the directory for our mailserver. These are :

- c:\inetpub\ftproot
- c:\inetpub\mailroot

Problem with those two directories is that the ACL is by default FULL CONTROL for Everyone. It should be put a little bit tighter, but which does provide the desired level of functionality. It's also best to use disk-quotas if you are going to make this directory writable to users. Otherwise a Denial-of-Service attack can be done against your system by flooding the system-drive.

Changing the ACL on the IIS Logfile

Logfiles under IIS are saved by default in %systemroot%\system32\LogFiles. You should make sure the ACL on this directory is set to:

- Full control for administrators
- Full control for system
- RWC for everyone.

Enable Logging

Logging is a necessity. If you do not keep any logs, detailed logs, you will have one hell of a job in front of you as soon as your system gets compromised. For IIS, W3C Extended Logging is advised. You activate it using the following procedure:

1. Load the Internet Information Services Tool
2. Right-click on your site (standard "Default Website"), choose properties
3. Click on the web-site tab
4. Check "Enable Logging"
5. Choose W3C Extended Log File Format as your Active Log Format
6. Click "Properties"
7. Now set the following properties on the "Extended Properties"-Tab:

- Client IP Adress
- Username
- Method
- URI Stem
- HTTP Status
- Win32 Status
- User Agent
- Server IP
- Server Port

The last two are only necessary if you run multiple websites on your site. The Status-field can come in very handy as debugging info on system crashes.

Set IP/DNS-restrictions

You can restrict access to users of a certain IP-range or domain by using IP / DNS-address restrictions. If you choose to use this option, we also advice to configure your packetfilter / IPSec PortFilter-policy in the same manner. It is better to deny service to the box at all than just to one service.

Please also take note of the fact that, if you enter DNS-names here, a DNS-lookup will be performed for every connect. This will be noticeable in a heavy decrease in performance. It is also less safe, since DNS can be spoofed... IP-adresses can be spoofed too, but it is quite hard to entirely spoof a complete connection instead of just a connection-attempt.

Securing the Metabase

The Metabase is a database which contains all the IIS-configuration parameters. It is located in the file Metabase.bin in the %SYSTEMROOT%\system32\inetsrv-directory. To obscure things for possible attackers, it is advised to hide this file from Unauthorized Users. To do this, stop the IIS-service, rename the file to a less visible filename,open the registry-key
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\inetmgr\Parameters
and add a REG_SZ-value "Metadata File" containing the complete path to the new Metabase-filename. Now restart IIS.

Move the website and FTP-root to another drive

Except when using a "quiet", scripted install of IIS, it is not possible to put the webroot and ftproot on another drive during installation. To minimize the risk however, this is a real necessity. Some known exploits to IIS only make it possible to traverse to the C-drive... putting the IIS-webroot on another partition makes defacing just a tad harder. After installation, you can make webroot-directories on another partition (with not too obvious names), preferably deeply nested within the directory-structure and with decent ACL's on them, and adjust the location of the webroot in the IIS Management Software.

Executable content should be validated first for trustworthiness

Before using executable programs as CGI-scripts, you should make sure that you understand the workings of these programs fully. It is best to run only self-developed applications on a secure webserver. Microsoft however has provided us with a way to see whether an executable does certain API-level calls. "DumpBin" is a tool included with many development-tools.

Usage speaks for itself actually... just perform something like this:

```
dumpbin /imports MyISAPI.dll | find "RevertToSelf"
```

If you get no feedback, the ISAPI does not make a direct call to "RevertToSelf". However there are other ways to perform API-calls, like the LoadLibrary-call. You should search for those calls too.

Update ROOT CA-certificates on IIS-server

Security is almost entirely built upon the understanding of "trust". To secure a machine, that machine should become a bit paranoid. It shouldn't be trusting the big bad world outhere. A great new improvement over the last couple of years has been the use of certificates, which prove to a third party that you are who you claim to be.

To make sure your machine can behave in this manner uninfluenced by "unclear" networks, it is advised to keep the database of Certificate Authorities on your webserver up-to-date. This includes:

- Add root-certificate authorities you trust
- Remove all root-certificate authorities which you do not know, or at least perform a background-check on them

You can reach all certificates using the MMC, Microsoft Management Console. Choose the Add/Remove snap-in (console menu) and click Add. Here you can choose to add certificates.

Remove all sample applications

As you most likely have noticed already, the process of hardening a server consists for the biggest part of removing services which you do not need. To show off the power of IIS, Microsoft included a lot of sample applications with its IIS-server. These are all located in Virtual directories, but are not installed by default. Delete them, cause they only constitute more risks when a hacker takes control of your system.

You can find some in \IISSAMPLES, \IISHELP and \MSADC. Especially the last one was quite notorious in 2000 due to the MSADC-bug. It was possible to use the RDS, Remote Data Services, a component of MDAC (Microsoft Data Access Component) to spawn a remote shell. You can read more about it in this security-bulletin.

Remove unneeded COM-components

Most COM-components are not useful for most applications and should be removed, cause they could become a security-risk. It is especially advised to remove the "File System Object"-component. This however is necessary if you are going to use Site Server 3.0. To disable the File System Object, execute this command:

regsvr32 scrrun.dll /u

Remove the IISADMPWD Virtual Directory

The IISADMPWD can be used to reset Windows NT & Win2k-passwords. It is designed for intranet scenarios, and should not even be in existance on plain IIS5-installs. However, when you upgrade from IIS4 to 5, this directory is available, and it is advised that it would be removed, if you do not require its functionality. You shouldn't use it on a publicly accessible webserver.

One of the problems with IISADMPWD was found on the 14th of may 2000, concerning a remote Denial-Of-Service against IISADMPWD on IIS 4 & 5. Malicious users were able to make the CPU-usage of inetinfo.exe rise as a rocket straight up to 100%. Inetinfo couldn't even be stopped, so a full reboot was necessary on IIS4. On IIS5, only the DLL that handled .htr-requests crashed, there was no system-wide DoS found there.

There also was this little "feature" where the IISADMPWD-directory contained a couple of vulnerable .htr-files from which it was possible to check if a certain account existed on the server.

Remove unused Script Mappings

IIS has filename-extensions predefined for lots and lots of filetypes, like asp & html. As soon as a request is received for one of these filetypes, a DLL is called with the extension, and code is sought to interpret the file. In some cases, like .idq, the handler gives "weird" error-codes, which might even reveal the install-path of IIS, or the path of the webroot. For this reason, it is advised to remove all filetypes which are not in use on your system.

You can do this in the Internet Services Manager, click on your webserver and choose properties.
Now choose **WWW Server**, **Edit**, **Homedirectory**, **Configuration**
At the very least, remove these extensions:

- **.htr** : web-based password reset
- **.idc** : Internet Database-connector
- **.stm,.shtm,.shtml** : Server-side includes
- **.printer** : Internet printing (BUFFER OVERFLOW on IIS5)
- **.htw,.ida,.idq**: Index server (REVEAL WEBROOT on IIS4/5)

The .printer extension really has a tale of its own to tell, which was ignored by many system administrators way back when this bug was really floroushing. It can also be added/removed in your group policy, and this setting overrides your IIS5-configuration. Since many administrators removed the mapping in IIS... they often forget about the group policy, and their webservers are still open to compromise.

Check your ASP-code

ASP, *Advanced Server Pages* opens up a lot of possibilities for beautiful things you can do in building entirely dynamic sites. However, broken ASP-code can also break the security of your webserver. Therefor it is advised to check, double, and even triple-check all ASP-code you use on your site.

It is especially important to check all user-input you get, since often ASP-code just throws the user-received input into an SQL-statement. By using commands to escape from the string-field, it becomes possible to execute arbitrary commands.

Disable parent paths

It's also very important to disable the use of ".." in function-calls like MapPath. This option is enabled by default, but you should disable it.

- Right-click your webroot, choose properties
- Click the "Home-directory"-tab
- Click on Configuration
- Click on App Options
- Uncheck 'Enable Parent Paths' checkbox

Disable IP-adress in content-location

The Content-location HTTP-header can expose internal ip-adresses. Like we said previously, security is partly based on topology, so it is important to keep your topology as obscure as possible.

To achieve this, we would like to have the internal ip-adress replaced by the FQDN (Fully Qualified Domain Name) in the Content-Location Header.

The way to do this is quite different for IIS4 and IIS5, so we will mention them here both separately.

<u>Before we start however, first a little warning from Microsoft</u>
"**WARNING:** *Using the Adsutil.vbs file incorrectly causes serious problems that requires you to reinstall Internet Information Server 4.0. Microsoft cannot guarantee that problems resulting from the incorrect use of the Adsutil.vbs file can be solved. Use the Adsutil.vbs file at your own risk.*"
It works fine at our systems though...

IIS 4.0

- Open a command-windows (start->run->cmd.exe)
- cd %systemroot%\system32\inetsrv\adminsamples
- adsutil set w3svc/UseHostName True
- net stop iisadmin /y
- net start w3svc

IIS 5.0

- Open a command-window (start->run->cmd.exe)
- cd inetpub\adminscripts
- adsutil set w3svc/UseHostName True
- net stop iisadmin /y
- net start w3svc

There is also a small workaround, for if the above is not possible... due to one or the other reason (you deleted the adminscripts previously,...). If you parse all HTML through the ASP-engine and create a custom header, you can totally rewrite the Content-location header.

You can implement this by following these guidelines:

- Rename all .htm/.html pages to .asp.
- Start the Internet Service Manager and load the IIS Snap-in for the Microsoft Management Console
- Under Internet Information Server, right click "default web site", and click properties
- Click on the HTTP-headers-tab.
- Choose "add" to Custom HTTP Headers.
- Type "Content Location" in the Custom Header Name (no "'s ;)).
- Type "http://www.mydomain.com" or whatever you want as Location Header in "Custom Header Value".
- Click OK twice.

Shielding your webserver

One of the best ways to protect a webserver is by having every request filtered before it is being passed to the server itself. There are two commonly used ways of doing this.

First of all, it is possible to introduce an ISAPI filter into the IIS environment which checks every request and passes it on to the webserver. This is the method used by Microsoft's **URLScan**. It has some major drawbacks, e.g. it can't protect any other webserver than Microsoft's IIS, and since it still runs in the environment of the webserver itself (the same machine), there is still a possibility of an overflow of the filter, which would also immediately grant shell-access. Also, it does not "analyse" input given to the webserver, but just passes or rejects it based on hard-to-configure rulesets. Less important, but also a drawback is the fact that its output is not configurable.

The second, preferred method, is the use of a reverse proxy which is being run in front of the webserver. This type of security is applied by e.g. **Ubizen's DMZ/Shield** or the opensource **TwHttpd**-package. The proxy is run on the normal ip-adress of the webserver, on port 80, while the webserver itself is moved to a higher port which is blocked from the outside. All requests are sent to the proxy, which checks them for validity, whether there are characters in the string that signify the possibility of shellcode or other malicious activity, and when everything looks quite peaceful, passes it on to the real webserver. In this case, the proxy can be run on another host than the webserver, which increases performance, security and availability (you can use load-balancing for the filter itself). Also, the filter can protect multiple brands of webservers and even multiple webservers at the same time. DMZ/Shield runs on Solaris and TwHTTPD is available for the Linux-platform.

Finalizing Security Settings

To finalize security-settings, there are some small things which can be done to make your system more resistant against DoS-attacks. With IIS, you have the option to do Bandwidth Throttling. This may be compared in a limited way with QoS (Quality of Service) and traffic shaping, only this does not influence or takes into account other services. With Bandwidth Throttling you can specify a max bandwidth which your IIS-server is allowed to use, in kb/s. I have this set to 512kb/s, which is about half my available bandwidth.

You can find this option here:

- Start, Control Panel, Administrative Services, Internet Information Services
- Right-click on your website
- The option can be found in the page "Performance"

In the same menu we find the "Enable Process Throttling" option. With this option you can decide how much the maximum CPU-usage is that the IIS-process can use on your server. If you wish to enforce the limit, also select the "Enforce limits"-checkbox. This option can prevent Denial-of-Service attacks.

To finish the secure setup, another interesting option can be found in the "Home

Directory"-window. This only applies if you really wish to host websites on the box, not just redirect the server to another website. First, make sure that the settings for your webroot are correct. Most of the time you do not want users to view script-code, just execute it. So, disable "Script source access" in the path-window. A bit more interesting are the Application-settings. If you do not run any scripts at all on your site, select "None", at the "Execute permissions" drop-down box. It is also advised to run these applications in "High (Isolated)" Application-protection mode.

Behind the "Configuration"-button, in the "App Debugging" window, disable "Send detailed ASP error messages to client, as to reveal as little as possible from your local setup. Disable all types of script-debugging.

It is also very important to stay up-to-date with all hotfixes for IIS... A standard install is vulnerable to the unicode-bug, IIS 4 even to RDS / MSADC. When you activated all security measures we spoke about in this step-by-step tutorial, none of these bugs will have a large impact. However, keeping your server bugfree is also very important. If you have not yet patched old errors, try installing the MegaPatch which Microsoft released in may 2001. It includes upgrades which fix all known directory-traversals and remote execution bugs found since Unicode. You can download it at Microsoft.com/security.

Chapter 4 NT Penetration

4.1 The Scan

When penetrating a system, the first thing an attacker will do is build an image of your host, as complete as possible. At first, the attacker may use some kind of portscanning-tool like Nmap to determine which ports are open on your system, or connect to your system anonymously to enumerate information (null-sessions), possibly to be used in social engineering-attacks.

4.1.1 The Portscan

The result of such a scan looks a bit like this:

```
[maarten@daemon tools]# nmap -O -P0 winbox.daemon.be

Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Interesting ports on winbox.daemon.be (192.168.0.5):
(The 1509 ports scanned but not shown below are in state: closed)
Port State Service
25/tcp open smtp
80/tcp open http
110/tcp open pop-3
135/tcp open loc-srv
143/tcp open imap2
443/tcp open https
1025/tcp open listen
1026/tcp open nterm
6666/tcp filtered irc-serv
6667/tcp filtered irc
6668/tcp filtered irc
7000/tcp filtered afs3-fileserver
8080/tcp filtered http-proxy

TCP Sequence Prediction: Class=random positive increments
Difficulty=11462 (Worthy challenge)
Remote operating system guess: Windows 2000 RC1 through final release

Nmap run completed -- 1 IP address (1 host up) scanned in 2 seconds
[maarten@daemon tools]#
```

In this example I have scanned a host (winbox.daemon.be) from a Linux-machine... It shows us which ports are open on the machine and which ports are not. "Filtered" means that a firewall, filter, or other network obstacle is covering the port and prevents nmap from determining whether the port is open.

Now, what can we do with this data? For example, lets say ports 23 (telnet), 22 (ssh), 21 (ftp) and 80 (http) are open. Now the easiest thing to do is a little banner-grabbing. You connect to the port using telnet, and look at the banner. Does it tell you something about the version of the daemon running?
On the host which was scanned above, we can for example check the version of IIS it is running, or maybe some other kind of http-daemon?

For example:

```
[maarten@daemon tools]$ telnet winbox.daemon.be 80
Trying 192.168.0.5...
Connected to winbox.daemon.be.
Escape character is '^]'.
GET / HTTP/1.1
Host: winbox.daemon.be

HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Content-Location: http://winbox.daemon.be/index.html
Date: Sat, 30 Jun 2001 10:46:29 GMT
```

This gives a clear picture on the brand of webserver. In this case it is IIS 5.0. Now, the regular script-kiddie will visit his favorite exploit-library and look for one which can be used against your system. If not, he might go through another proces to identify the software you are running (passive network mapping, queso, os fingerprinting, nmap -O). When no vulnerable service is found, he can download the sourcecode of one of your daemons (if opensource) and go bughunting himself, or can try to find some misconfiguration on the box.

4.1.2 Null-sessions

Null-sessions are NETBIOS-connections which occur without a password. A user logs on to your machine using a blank password. When this occurs, the user will not be able to access any password-protected shares, but will be able to enumerate information from your system, like, for example, share-names, usernames, registry-information.

The reason that these null-sessions exist is the fact that untrusted machines, located on other domains might also require some information. A domain controller from another domain might gather information about the shares.

If you run Netbios only within your local domain, disabling null-sessions, as described in the Hardening NT & Hardening Win2k-part, will form no problem at all. All users will make valid password-protected connections to your domain-controller and will be able to succesfully authenticate and retrieve information. Users from external, trusted and untrusted domains will however have problems accessing shares on your domain. Nothing can be done about this.

Another way to disable null-sessions, or minimize their impact, is to use the NTFS-permissions to give null-sessions the ability to enumerate shares, but do not give it access to files. Null-sessions belong to the group "Everyone", not to "Authenticated Users". Everywhere NTFS-permissions have group "Everyone"-permissions, the null-session will have permissions. Null-sessions cannot belong to the group "Authenticated Users".

The Null-session problem cannot be solved easily. Best is to block all access to netbios-ports from untrusted networks, like your internet-uplink, using a packet-filter.

4.2 The Exploit

Exploits answer *vulnerabilities*. The goal of an exploit is to obtain *higher privileges* than you already have. For example, when you can only browse a specific machine, an exploit *could* take care of giving you a shell on that system. If you already have a shell, its function might be to give you administrator-privileges.

Most exploits are written purely as proof-of-concept, to show that a certain vulnerability *can* be exploited to do something on the system the real admin wouldn't really want you to do. But they get abused too.

Exploits aren't always necessary coded, they can also be a way of thinking, a way of doing things. For example, let's suppose that winbox.daemon.be is a non-patched Win2000 Advanced Server standard-install. When I surf to this *http://winbox.daemon.be/scripts/..%c0%af../winnt/system32/cmd.exe?+/c+dir*, chances are high that I will get a directory-listing (the so-called unicode-bug, or unicode directory-traversal vulnerability). We aren't using exploit-code, but we are exploiting a known issue in this webserver.

Exploits can be found in many places on the web, and are often used with malicious intent. The real meaning of exploit-code however is good, we use it to demonstrate security-issues on a certain host. We're not here to exploit them, we're here to stop others from exploiting them...

Now, on winbox.daemon.be (win2k), which has not been hardened, you will only be able to run commands on the IUSR_winbox-account. With some thinking and work this can be expanded to administrator-access, ofcourse.

Another, even more serious flaw in IIS 5 is for example the ISAPI-.printer-buffer overflow. You overflow a buffer on the .printer-extension on IIS, which executes some code on the stack, so-called shellcode. A malicious hacker might use a well-known exploit, jill.c for this. He runs it, gets administrator-access, and from this moment, your box is lost. As soon as it was penetrated up to administrator-level, it is almost *impossible* to make sure that you removed all backdoors from the system, and a full reinstall will be the fastest way to get your server back up.

4.3 Shellcode

We already talked about shellcode previously. Shellcode is code that gets executed on the remote system, through the exploit.

Shellcode can do one of many things:

- Upload an executable to the webroot (making it executable by http-request)
- Enable a Guest-user, add it to the administrators-group
- Download an executable and execute it
- Bind a command-shell to a TCP-port

and so on…

A bit more special is the *"Reverse Command Shell"* This code connects to a specific TCP port on a different remote machine, and binds a shell (cmd.exe) to it. For example the well-known *jill*-exploit used to exploit the ISAPI .printer-buffer overflow works this way. This comes in handy for attackers when they have root on a different box in the same network, and inbound connections to any other port than e.g. 80 are not allowed. Intrusion Detection Systems which only take inbound traffic into account are also fooled.

Due to the fact that shellcode ought to be as small and clean as possible, some hackers use so-called "chained shellcode". Using this shellcode-variant, it is perfectly possible to make your shellcode as big as you want. The only thing the shellcode does is create an environment for another piece of shellcode, connect to a machine running a self-written daemon, and fetch the larger shellcode. This is then executed in the self-created environment. Using chained shellcode, it is possible to avoid, e.g., a renamed cmd.exe, because you have enough space to execute your own command interpreter (long shot, haven't seen this in the wild yet), or at least enough to go looking for a useable environment. Almost all other types of shellcode require a valid cmd.exe.

The most "underground" and least used type of shellcode is "recycling shellcode". With this type of shellcode you can evade even the most tight firewall-protection. The shellcode spawns a command-shell over exactly the same TCP-connection that was used for exploitation. Let's look for example at a box that ONLY allows port 80 in, port 80 out (due to firewalling etc...). You cannot bind anything to port 80, cause the webserver is already running at that port. So, you hijack the http-connection that was used to exploit IIS, and spawn a command-shell through exactly the same connection.

4.4 Obtaining Administrator-privileges

Due to the fact that the whole suid-mechanism isn't as cleanly implemented on NT as on Unix, NT is often seen as more secure to local attacks... however, this is caused by the fact that NT just isn't used as a shell-operating system.

When attackers are able to obtain a user-shell, for example through misuse of the unicode flaw in IIS4/5, they most likely want to obtain administrator-privileges. There are some ways to do this, the oldest being a small program called "GetAdmin", which would just change the global NT User-Flag, causing the function NTOpenProcess not to check for privileges when launching a program. This only works on old NT-servers, which you probably are not running, so this problem has faded over the years.

Most of the time however, becoming an administrator depends on good old intuition. A hacker looks for scripts, programs, batch-files that are being executed by the administrator, and which they can write to. Then they add commands to the batch-files or replace the program by a piece of code that does what they want it to do. For example, many NT-administrators write a couple of backup-scripts, don't think about putting an ACL on those files, and plan those scripts to be run every night. Every planned event on WinNT / Win2k is being executed under the privileges of LocalSystem, which is even better than administrator-privileges. Hackers who obtain a shell on the system can just add commands to these backup-scripts, and create user-accounts, delete system information, deface your website, everything what a normal administrator can do.

To prevent this from happening, you should take care of EVERY single file on your system, check its ACLs, audit it if it is important... There is a tool which might come in very handy, called DumpACL, which dumps all ACL's to a file for later inspection. You can obtain it at [http://www.somarsoft.com].

4.5 Going in deeper... ring 0

When you're an administrator... you still can't do everything you might want. Being compliant to the TCSEC (Trusted Computer System Evaluation Criteria), NT has all of its security-relevant functions in one, highly audited part of the operating system, called a TCB, Trusted Computing Base. This TCB normally can and may not be altered in any way. When it is changed, you do have some serious problems.

When your TCB is touched, people can evade ALL kinds of host-based intrusion-detection. All IDS's trust on the operating system to give correct information, which is necessary to access NTFS-files. When the TCB is owned by a hacker, he can make it look like all checksums are ok, he can make network-traffic invisible to all host-based packetfilters (imagine having this problem on a corporate firewall).

The TCB should obviously be well protected. On NT however, mostly due to the nature of it's closed-source, there are multiple ways to abuse the TCB...

Most of the time, this is done by inserting a malicious kernel-mode driver. However, to insert such a driver into the running kernel, it is always necessary for the user to give his clearance to install this driver. You can mask it to look like something else, for example a printer-driver. The inserted code can then run on level 0, in the Trusted Computing Base.

Another way is abusing fawl-entries to the kernel. It is proven, but most of the time classified, that there are ways to get code into the kernel without the TCB even noticing it... One is based on a bug in the SystemLoadImage-api-call. When abusing such a method, you can insert your own code into the kernel, compromising the entire system.

4.6 Rootkits

Rootkits are pieces of software that are run at kernel-level, in the Trusted Computing Base, which have the ability to backdoor the entire system. They can offer alternate ways in, they can provide sniffing or snooping-facilities, they can violate security in a very strange way, cause all ACL's will still be active, but they will be entirely ignored by the system.

Currently, the most popular rootkit for Windows NT is being developed by Greg Hoglund, at [www.rootkit.com]. The driver is inserted as a kernel-level driver, and deletes all Access-Control and all security-checks in the TCB. If you dive into your security-settings, you will notice they are all in place, but that any user can delete any file and execute any program.

Currently, the only way to prevent your system from being "infected" with a rootkit is installing a piece of software called IPD, the *Integrity Protection Driver*. This piece of software uses the same techniques as rootkits, but instead locks these techniques down so it becomes impossible to insert code these ways. It effectively "blocks" Windows's ability to insert new drivers by anyone. It does this twenty minutes after bootup, so all new hardware-drivers you install will need to be installed within 20 minutes after boot. Removing the driver is also only possible within those first twenty minutes. Since this driver breaks a whole lot of the windows-functionality, it is not advised for workstations, but it is a nice solution for webservers, application-servers which only need to run a small set of selected software. You can get the IPD at [http://www.pedestalsoftware.com/intact/ipd/]. Read more about it in chapter 5.2

Chapter 5 Tools

5.1 Replacing Telnet by SSH

Telnet is a popular way to do remote management on Win32-boxes. The protocol itself however, is extremely insecure. When you make clean telnet-connections, all information, including usernames and passwords cross the wire in plaintext. This offers the ability to malicious users on your network to sniff the network-traffic and thus steal your passwords. It is also more likely for a user to hijack your connection, as he has a perfectly clear view at the state which your connection is in.

The problem with cleartext-passwords was, for a part at-least, addressed in NT by the use of NTLM-Authentification. However, it still is more secure to encrypt your entire connection, especially with the large quantity of bugs which were recently discovered in the Microsoft Telnet Daemon, and the fact that even when NTLM is used, the data itself crosses the wire in cleartext, so remains sniffable.

If you would like to run an SSH-daemon on a Win32-box, you have two options. Either you can use the opensource port from Unix, or you can use a commercial package. In this chapter I will discuss and illustrate the installation of the Unix-port, since it is free and offers the same functionality as any commercial package.

The required items for this setup

To be able to run the Unix-SSH port, you will require the following items:

- Cygnus / GNU-Win32
  `Get it at http://www.cygnus.com`

Installing SSHD

The actual installation of the SSH-daemon doesn't take long at all. First of all, install the Cygwin-package. Next, it is important to add the main-CYGWIN-directories to your PATH-environment variable.

For example, if you installed cygwin and you have a d:\bin-directory, execute this command:

```
set path = %path%;d:\bin;d:\bin\ssh
```

Another environment variable which is of large importance is the CYGWIN32-variable.

```
set CYGWIN32=tty
```

Next, you will have to create your passwords and SSH-key.

```
mkpasswd -l > \etc\passwd
ssh-keygen1 -f /etc/ssh_host_key
```

When you let the SSH-daemon log on users, you will have to give it appropriate rights to do this. For this, make a new usergroup for example "sshusers". Add all users to this group, which you want to be able to log in using SSH.

Now, open up the Policy Management Snap-in. Give the group "sshusers" these rights:

- Act as part of the Operating System
- Increase Quotas
- Replace a process-level token

Next, run /bin/sshd and your SSH-daemon should be up and running. If you wish, it is also possible to run SSHD as a service on your system. To do this, you will need the Windows NT Resource Kit, especially the "instsrv" and "srvany"-tools. Again, I expect you to have the sshd.exe-binary in d:\bin. Your NT Resource Kit-directory is c:\ntreskit in this example.

Execute: *c:\ntreskit\instsrv SSHD "c:\ntreskit\srvany.exe"*

Now the service SSHD has been added to the list of available Microsoft Windows-services. Everytime this service starts it will try to load c:\ntreskit\srvany.exe. Now, the only thing that rests us is to give srvany.exe its parameters.

To do this, run REGEDIT.EXE (Start/Run/REGEDIT), and add the key "Parameters" under HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SSHD
Now, enter "d:\bin\sshd" as the value for this new key.

If you experience any problems, which often happens on the Windows 2000 series, you can try to run sshd with "-v" or "-d" as parameter. "-v" stands for "verbose"-mode and gives lots of extra status-info. "-d" runs sshd in debug-mode. After the first connection it will exit, and give you a review of what happened.

Please note that SSH/SSL is not a perfect solution to the problem of encrypting network-traffic. Some holes have been found in different versions of the SSH/SSL-protocol, most of which are limited to the release of some data and making it possible for man-in-the-middle attacks to occur. For example, when you SSH into your Windows-NT-box, and sniffing is possible, it might be possible for attackers to retrieve some information like the length of your password and the length of the commands you enter. This issue was raised by Solar Designer and Dug Song, for example at HAL2001 in the Netherlands. However, it is one of the first steps you should take in securing your network. Defense should be built in layers, in which all layers have a specific function, and each keep out unwanted visitors.

5.2 Implementing IPD to prevent the installation of rootkits

Little after Greg Hoglund released his Windows NT Rootkit, people started thinking of solutions. One of the most popular solutions is to install a device-driver on the system, which prevents all ways to install drivers on the system, and blocks some of the common ways rootkits might get into the Trusted Computing Base. The largest disadvantage of this system is the fact that once your "anti-rootkit" is in place, it will become impossible for example to install a new service, to install a printer-driver, ... Another problem is that only known holes can be fixed. If someone finds another covert path into the TCB, it isn't that hard at all to bypass such systems.

The most popular of the anti-rootkit-tools is called the IPD, or "Integrity Protection Driver" and is released as freeware by Pedestalsoftware [http://www.pedestalsoftware.com].

First of all, the IPD blocks write-access to the part of the registry concerning services, it secures the subdirectories op winnt\system32, and restrects some things to all parts of the system, except some important components. It also blocks a couple of systemcalls which can be used to import kernel-level drivers.

The IPD is loaded as a service on bootup, but only activates 20 minutes after boot. This is done to make it possible to disable the driver. Normally, this can't be done, because the driver even blocks access to its own registry-entries, but, within the 20 first minutes after boot you can disable it. Also, if you would like to install a printer or other device, this can be done within that time-limit.

For best results, it is advised to start a webserver running the IPD, and only enable its network-uplink after 25 minutes, to make sure the system integrity is protected by the IPD.

Getting the IPD

You can download the IPD at http://www.pedestalsoftware.com/download/ipd.zip .

Installing the IPD

Unzip the zip-file you just downloaded in a random directory. Then run the ipdinstall.exe program like this:

ipdinstall.exe install

The driver is now installed. Wait twenty minutes for activation. You will not have to repeat this procedure as the driver is added to your services-list and will automatically be activated at next bootup.

Removing the IPD

In case you want to disable the IPD, you will have to do so while the driver is not activated, so within 20 minutes after bootup. Remember that it is impossible to remove the driver using the services-panel of Windows NT / 2000.

To disable:

Ipdinstall.exe remove

5.3 Running all TCP-connections SSL-encapsulated

In the previous chapter we already discussed the need to do remote management purely over secure connections. Another problem that meets the light of day when we are talking about security, is the fact that a system administrator, or any user for that matter, often uses the same password on clear-text as on encrypted services. If a malicious user obtains the password from, let's say, sniffing a POP3-connection, your network is still as wide open as it gets.

For this reason, all possible TCP-connections should get encrypted before any data hits the wire. In this example we will be discussing the use of STUNNEL, a piece of majestic opensource software, and the encryption of POP3 (incoming mail)-connections.

First of all, you will require a mailserver that accepts SSL-connections. Normally, the port-number of POP3 is 110. This is for all unencrypted connections. Encrypted POP3, also called "SPOP3"for Secure POP3, has 995 as its standard-portnumber. If your mailserver itself supports SPOP3-connections, all of this is not necessary and you can, most of the time, just setup your client to retrieve your mail in a secure fashion.

Getting the tools

- STUNNEL-3.14.exe (or more recent) from http://www.stunnel.org/download/binaries.html
- STUNNEL.pem from http://www.stunnel.org/download/stunnel/win32/stunnel.pem
- libssl32.dll and libeay32.dll from http://www.stunnel.org/download/binaries.html
  `Copy these to c:\winnt\system32`

Encrypting your link

On your mailserver, run STUNNEL like this:

```
stunnel-3.14 -d 995 -r localhost:110
```

Now, every connection coming in on port 995 will be decrypted and sent in cleartext to localhost:110. All unencrypted traffic is now running on localhost, so the passwords cannot be sniffed on remote hosts. You now only have to make sure all your clients are setup to connect with SSL on port 995. If your clients don't support SSL-connections, run stunnel on each of them like this:

```
stunnel-3.14 -d 2500 -r mailserver:995 -c
```

Stunnel opens local port 2500, and forwards it ssl-secured to port 995 of your mailserver (replace "mailserver" with the hostname). You now setup your mailclient to connect to localhost:2500 instead of your ordinary mailserver.

5.4 Using the Microsoft Management Console

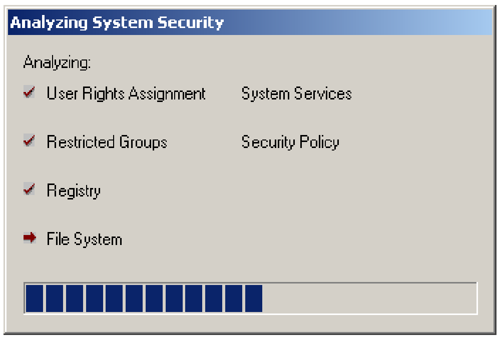
The Microsoft Management Console is quite a handy application to better build up a basic security of NT and Windows 2000 systems. For Windows NT, you will have to download it by hand and install it yourself on your system. On NT, you can use Microsoft Management Console 1.2 only if you are using Internet Explorer 5 or higher, which I would recommend against. Using IE5 on a production-NT-server is not advised. Version 1.1 works together with IE 4.01 SP2, which is a correct setup. On Windows 2000, it does not matter which version of either IE or the MMC you use.

On NT, you can find the Management Console in the Option Pack. On Win2k just click on "start", "run" and enter "mmc". Now click on "console", and select "Add/Remove snap-in". Select "Add", and from the list you get, select the "Security Analysis and Configuration" & "Security Templates"-subsystems. Close the selection-window and affirm your decision. Now go back to the main window of the Microsoft Management Console.

The Microsoft Management Console can be used to import security-templates in your system, compare your current setup with one of those templates, and apply one of those templates, changed or not, onto your current configuration. This way you can give your system basic security by just executing a single command. Fine-tuning afterwards however is still essential to any good security-policy. Something is not secure if the administrator does not know what has changed.

Right-click on "security analysis & configuration" and select "open database". As database-name you just enter the name of your server, for example "banshee" in my case. A new database "banshee.sdb" will be created. Now you will be asked which security-template you wish to open. For all kinds of server-setups templates have already been generated. The default (after-install) settings of the operatingsystem can be found in the basic*-templates. "Secure" setups can be found in the secure*-templates. Hi-security is offered by hisec*. "dc" stands for Domain-controller, both Primary and Backup, "ws" for Workstation / Server, "sv" for server and "ws" for workstation.

As soon as you open a template, the server will load it. Again, right-click on "security analysis and configuration", and select "Analyze Computer Now". Answer all questions asked, and the computer will commence a full analysis of your windows-system. You get a report in return with all security-related settings on your machine, and how they differ from the settings loaded into the template.



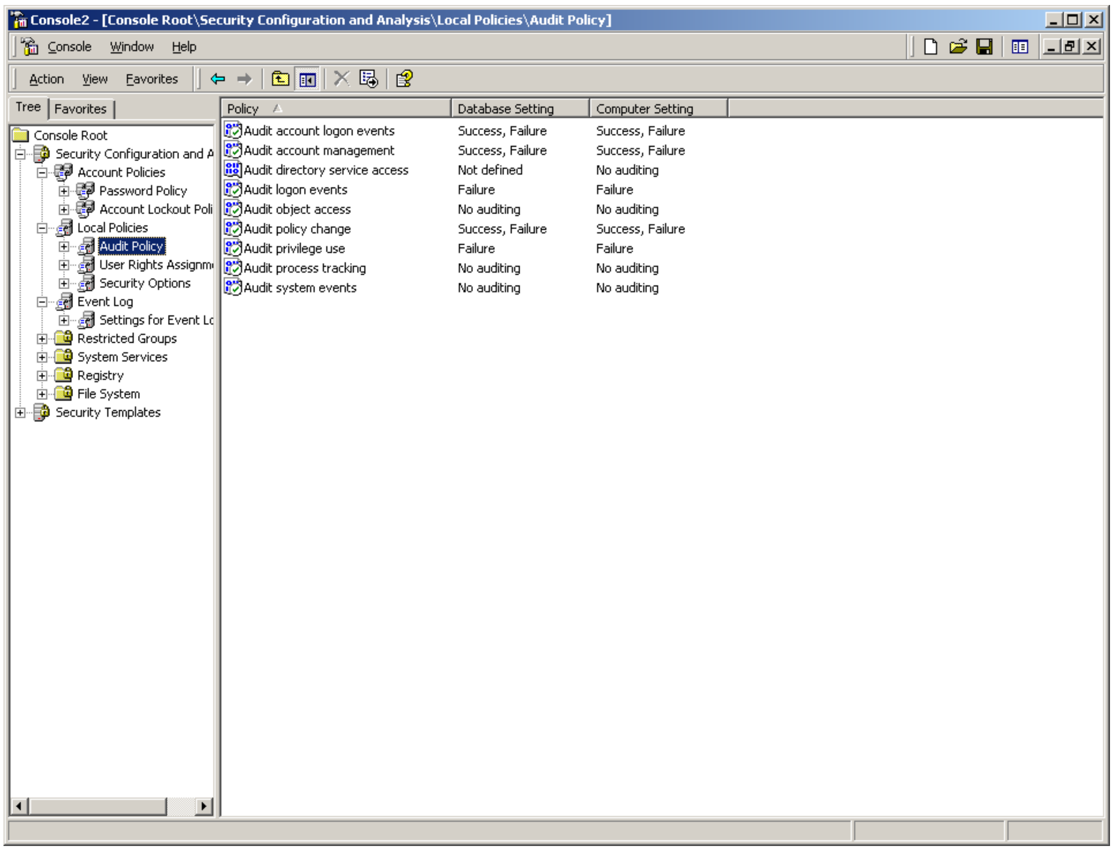
This is the place to start making changes to your configuration. Adjust policy-settings, change password-lockouts, services started. As you see, you have a handy control-panel which you can use to put all security-settings the way you want them, and you can choose a safe starting-point, secure, highly-secure, or just basic settings (through the template-selection).

After you have made the necessary changes, you will be able to apply them by right-clicking one more time on "security analysis and configuration", and choosing "Configure Computer Now". You might want to save your altered template to something you will be able to reuse on multiple machines.

The most interesting part is that the security editor of the MMC can also be used on the command-line, for example to integrate it into a batch-file to standardize a local security-policy on all machines. Prepare your correct policy on a sample-machine, and then distribute it to all machines, running, for example

```
SECEDIT /CONFIGURE /DB banshee.sdb /CFG dsec.cfg /LOG dsec.cfg /VERBOSE
```

To apply the policy to the local machine. If you put the policy on a network-drive, and install SECEDIT on every machine, this is a nice way of automating things.



Policy	Database Setting	Computer Setting
Audit account logon events	Success, Failure	Success, Failure
Audit account management	Success, Failure	Success, Failure
Audit directory service access	Not defined	No auditing
Audit logon events	Failure	Failure
Audit object access	No auditing	No auditing
Audit policy change	Success, Failure	Success, Failure
Audit privilege use	Failure	Failure
Audit process tracking	No auditing	No auditing
Audit system events	No auditing	No auditing

5.5 Auditing with NTLast

Went a bit overboard with logging? No problem... oh, you got rooted... big problem. You can go through tons and tons of log-printouts, or, you can use some kind of automated auditing-tool.

NTLast probably is one of the most well known programs that achieve this goal in let's call it an "elegant" way. NTLast is a nice little program, written by NTObjectives, recently acquired by FoundStone Security. With NTLast, you can cross-reference logged on users to specific problems. It works a whole lot more flexible than the standard NT / Win2k Event-viewer.

First of all, make sure that you are indeed auditing logons & logoffs from users. After you are certain, get NTLast from www.foundstone.com and extract the contents of the zipfile to a directory (you might want to put it in the secure path we are using for command-line-tools, remember?).

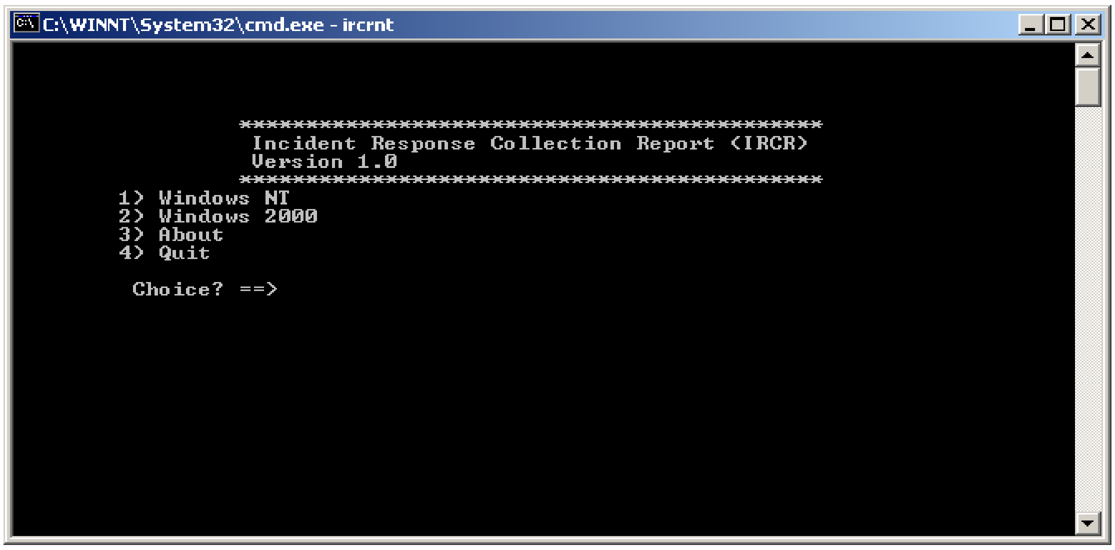
Next, you can run NTLast with a couple of different parameters. At the time of writing, version 3.0 is in wide circulation. A nice feature is that NTLast can be used to see the difference between local and remote logons. Using the "-r"-parameter, the last 10 remote logons are shown. Be sure to check out what this great piece of software can do.

5.6 Forensics Tools for NT

IRCR Incident Response Tools

The IRCR-tools is a nice program written by John Mcleod. The programs make an image of the system with most of the necessary forensics-info, so any user unfamiliar to the system can mail a blueprint to an experienced forensics-engineer.

Download the zip-file at http://www.incident-response.org/windowstools/IRCR.zip , unzip the file and extract all files to a directory. Then take a blank floppy, preferably not made on the possibly compromised system, and run IRCRNT.



```
                *********************************************
                 Incident Response Collection Report (IRCR)
                 Version 1.0
                *********************************************
        1) Windows NT
        2) Windows 2000
        3) About
        4) Quit

         Choice? ==>
```

Now select the platform on which you are running the program. The system will now make a forensics-blueprint and will write it to the floppy-disk. This will take 2-8 minutes on an average system, according to the author. On mine, it took 12, so, it depends a bit on the size of your harddrives.

After this procedure, you will find a ZIP-file on the floppydisk, containing a thorough dump of the application-log (applog.txt), in this format:

```
--------------------------------------------------------------------------
RecordNumber:   285
Source:         MSDTC
Computer:       BANSHEE
Category:       1
Event ID:       4097
EventType:      4
Time Generated: Sat Jul 28 14:04:24 2001
Time Written:   Sat Jul 28 14:04:24 2001
User:
Message: MS DTC has started.
```

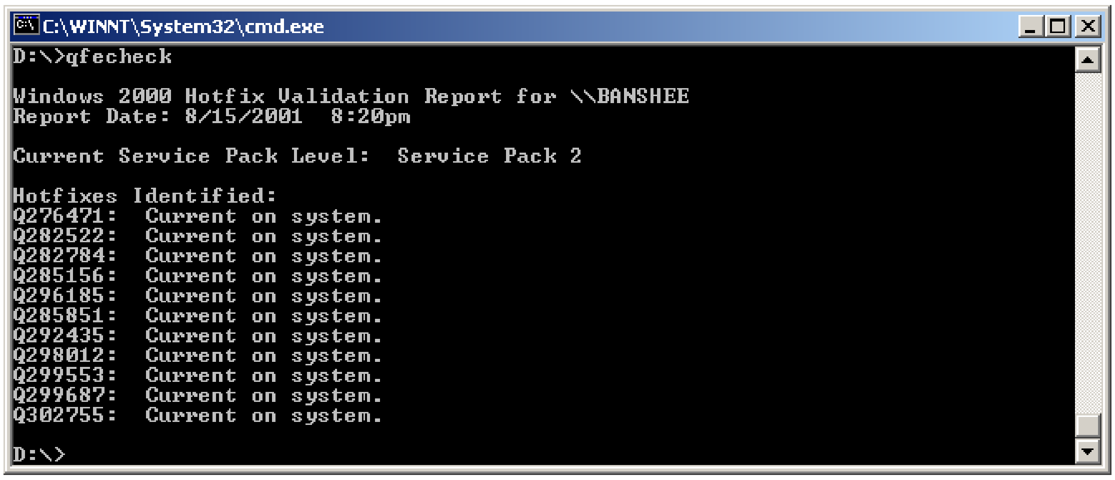
The event-log, a dump of the routing-table, the ARP-table, a list of all files on the system, and so much more. When correctly analyzed, these files offer a high quantity of information, which can lead to fruitful results.

5.7 Protection against Buffer Overflows

5.8 Microsoft Tools for ensuring security

QFECheck

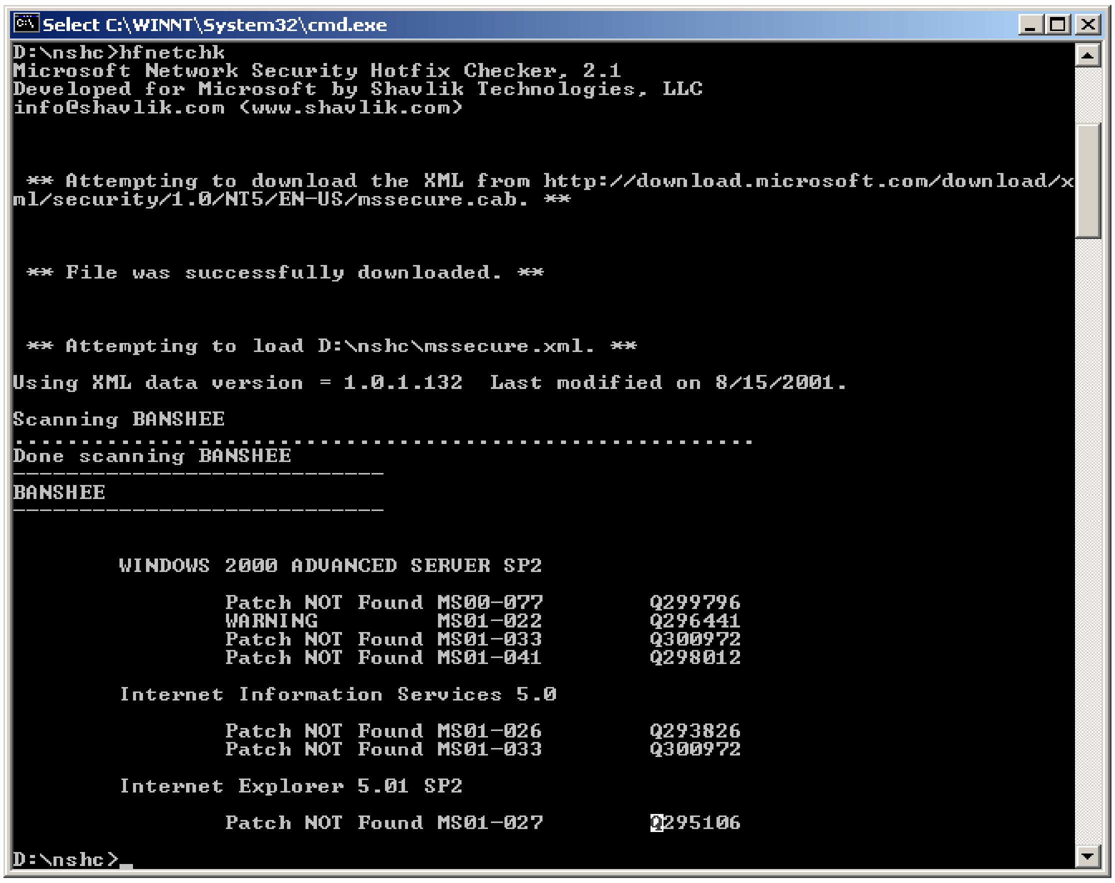
QFECheck is a small microsoft-program for Windows 2000 which allows a system administrator to take a look at which hotfixes are properly installed on the system. This is yet another way to make sure that your entire network is up to date. It gives you a list of which hotfixes are installed, are not installed, and are invalid, because for example one of the files have been overwritten. You can download the little tool @ Microsoft Security on this url:
http://support.microsoft.com/support/kb/articles/Q282/7/84.ASP



```
D:\>qfecheck

Windows 2000 Hotfix Validation Report for \\BANSHEE
Report Date: 8/15/2001  8:20pm

Current Service Pack Level:  Service Pack 2

Hotfixes Identified:
Q276471:  Current on system.
Q282522:  Current on system.
Q282784:  Current on system.
Q285156:  Current on system.
Q296185:  Current on system.
Q285851:  Current on system.
Q292435:  Current on system.
Q298012:  Current on system.
Q299553:  Current on system.
Q299687:  Current on system.
Q302755:  Current on system.

D:\>
```

NSHC

NSHC is yet another tool, developed by Shavlik Technologies for Microsoft, which does a check of the current system to see if all hotfixes have been applied.
You can get it @ http://download.microsoft.com/download/win2000platform/Utility/2.1/NT45/EN-US/nshc.exe



```
D:\nshc>hfnetchk
Microsoft Network Security Hotfix Checker, 2.1
Developed for Microsoft by Shavlik Technologies, LLC
info@shavlik.com (www.shavlik.com)


 ** Attempting to download the XML from http://download.microsoft.com/download/x
ml/security/1.0/NT5/EN-US/mssecure.cab. **


 ** File was successfully downloaded. **


 ** Attempting to load D:\nshc\mssecure.xml. **

Using XML data version = 1.0.1.132  Last modified on 8/15/2001.

Scanning BANSHEE
........................................................
Done scanning BANSHEE
----------------------------
BANSHEE
----------------------------


        WINDOWS 2000 ADVANCED SERVER SP2

                Patch NOT Found MS00-077        Q299796
                WARNING         MS01-022        Q296441
                Patch NOT Found MS01-033        Q300972
                Patch NOT Found MS01-041        Q298012

        Internet Information Services 5.0

                Patch NOT Found MS01-026        Q293826
                Patch NOT Found MS01-033        Q300972

        Internet Explorer 5.01 SP2

                Patch NOT Found MS01-027        Q295106

D:\nshc>_
```

Chapter 6 Hardening RRAS

6.1 Basic Guidelines

It is important to:

- Only allow the TCP/IP-protocol on the RAS-interface.
- Audit all RAS-connections.
- Centralize all RAS on a small amount, preferably one, server, this makes the entire RAS-setup auditable, and makes it easier to do maintenance. One NT-server supports up to 255 simultaneous RAS-connections. This might be a problem for availability when maintenance is needed.
- Do NOT allow users to build their own RAS-server on their desktops.

6.2 Selecting the correct Authentification Protocol for the job

In most cases there really isn't any choice, you have to use the protocol which is forced upon you by for example your provider, to login on an external server, or your clients have specific needs, to which you want to fulfill by giving them the protocol they want, to login to your RAS-server.

By default, Windows supports these types of authentification:

- PAP
- SPAP
- EAP
- CHAP & MS-CHAP

1. <u>PAP or "Password Authentification Protocol"</u>
   PAP probably is the most used protocol around, mostly due to its age, and due to the fact that it is one of the only protocols supported by most older operating systems. You probably used to use it to logon to your ISP by modem.
   However, PAP mostly has drawbacks besides the advantage of compatibility. All passwords are transmitted in cleartext, and can easily be sniffed. Also, passwords cannot be changed in the authentification-change, so users cannot change passwords themselves.

2. <u>SPAP or "Shiva Password Authentification Protocol"</u>
   SPAP is a special version of PAP designed by Shiva Technologies. It uses a (pretty weak) encryption algorithm, but the key to this encryption does not change. If someone sniffs the encrypted password, and builds a connection himself, retransmitting the encrypted password, it will be accepted and authentification is possible. This is called a *replay*-attack.

3. <u>EAP or "Extensible Authentification Protocol"</u>
   EAP offers some other methods of authentication for clients, like Kerberos, Certificates, and many more. It also allows for mutual authentification, so clients are also certain they have logged on to the correct host.

4. <u>CHAP & MS-CHAP or "Challenge Handshake Authentification Protocol"</u>
   CHAP is yet another step higher on the secure authentification-ladder. Using CHAP, two levels of encryption can be chosen, either DES can be used, or an MD5-hash will be used to encrypt authentification. The second is standard on NT. All passwords are uniquely encrypted, so replay-attacks are impossible.
   The authentification-phase looks a bit like this:
   - Server sends a challenge-string to the client
   - Client responds with a one-way encrypted value of that challenge-string
   - The server verifies this and authenticates the client

   Also, CHAP-connections are verified during the connection itself.

   Microsoft designed an expanded version of this standard, called MS-CHAP, which has as main advantage that it allows for a complete encryption of the connection using MPPE (when on a PPP or PPTP-connection). MPPE is a unique microsoft encryption protocol, described in RFC 3078. It uses RSA RC4-encryption with either 40 or 128-bit keys. Just like EAP, it allows for mutual authentification.

6.3 Configuring dialback

Whenever possible, you should think about using dialback for your clients. In this case, you can atleast have a certain degree of certainty that the correct host is logging on to your modempool.

When forging a connection, two things have to be forged, both the telephone-number of the user whose account is getting compromised will have to be obtained and tapped, and the (hopefully strong) passwords of the user will be required. This offers a reasonably high level of security, which used to be in general use, but nowadays has diminished due to the rise of internet-technology.

Using NT RAS, the modem itself does not need any dialback-functionality, since this is entirely done in software. The user dials in, authenticates, the line gets dropped, and the RAS Server will rebuild the connection instantaneously.

6.4 Secure inbound access with PPTP

When your RAS-clients support PPTP, use this technology to protect all network-traffic. PPTP, fully known as the "Point to Point Tunneling Protocol", is used to build VPN's or Virtual Private Networks, which are entirely encrypted tunnels of information. In this tunnel, you can use any protocol you want (in theory)... Currently supported are IPX, NetBEUI, NetBIOS and TCP/IP, although we highly advise against the use of any other protocol than TCP/IP on a PPTP-tunnel.

You can enable PPTP for inbound connections in the "Advanced" properties of the Dial-in-connection.

Chapter 7 TCSEC C2

What is a C2 Operating System?

A C2-Operating system is an operating system which has been certified by the research-department of the DoD to comply to a set of rules, which acknowledge whether a certain operating system is "secure", and in which degree, or is considered "insecure".

Windows NT 3.5 & 4.0 were both given the label "C2-compliant", considered as a stand-alone operating system. This does not mean that your server is C2-certified. Only installations can be "certified". The OS is only "compliant".

The complete TCSEC-review of Windows NT4 with Service Pack 6 can be found here:
http://www.radium.ncsc.mil/tpep/library/fers/TTAP-CSC-FER-99-001.pdf

C2 Operating systems fulfill the following demands:

- Discretionary Access Control
  "The Trusted Computing Base defines and controls access between users and objects in the computer system. The enforcement mechanism shall allow users to specify and control sharing of those objects between individuals or groups of individuals, or both, and shall provide controls to limit propagation of those access-rights. The discretionary access control mechanism shall, either by explicit user action or by default, provide that objects are protected from unauthorized access. These access controls shall be capable of including or excluding access to the granularity of a single user. Access permission to an object by users not already possessing access permission shall only be assigned by authorized users."

  In short, this means that Discretionary Access Control, from hereon mentioned to as DAC, is used to control access by assigning rights for users or groups of users to files. The owner of the file has the right to change these rights. This system is incorporated in Windows NT using the so-called Access Control Lists.

- Object Reuse
  "All authorizations to the information contained within a storage object shall be revoked prior to initial assignment, allocation or reallocation to a subject from the TCB's pool of unused storage objects. No information, including encrypted representations of information, produced by a prior subject's actions is to be available to any subject that obtains access to an object that has been released back to the system."

  When you use a bit of memory, and later on use the memory again, NT prevents the fact that any data stored in the memory is regained by reuse of that piece of memory.

- Accountability
  - Identification and Authentification

"The TCB shall require users to identify themselves to it before beginning to perform any other actions that the TCB is expected to mediate. Furthermore, the TCB shall use a protected mechanism (e.g., passwords) to authenticate the user's identity. The TCB shall protect authentication data so that it cannot be accessed by any unauthorized user. The TCB shall be able to enforce individual accountability by providing the capability to uniquely identify each individual ADP system user. The TCB shall also provide the capability of associating this identity with all auditable actions taken by that individual."

Every user has to authentificate via a secure way (like a password) to the system before he can do any actions performed by the TCB (getting access to files, …). Every user will be accountable for his or her actions.

- <u>Audit</u>
  "The TCB shall be able to create, maintain, and protect from modification or unauthorized access or destruction an audit trail of accesses to the objects it protects. The audit data shall be protected by the TCB so that read access to it is limited to those who are authorized for audit data. The TCB shall be able to record the following types of events: use of identification and authentication mechanisms, introduction or objects into a user's address space (e.g., file open, program initiation), deletion of objects, and actions taken by computer operators and system administrators and/or system security officers, and other security relevant events. For each recorded event, the audit record shall identify: date and time of the event, user, type of event, and success or failure of the event. For identification/authentication events the origin of request (e.g., terminal ID) shall be included in the audit record. For events that introduce an object into a user's address space and for object deletion events the audit record shall include the name of the object. The ADP system administrator shall be able to selectively audit the actions of any one or more users based on individual identity"

  The TCB must be able to create, maintain and protect modification of all audit-logs. Read-access should be limited to specific authorized users (for example the Security Administrator). Each recorded event shall be signed by:
  - Date and time
  - User
  - Type of event
  - Success/Failure
  - When an object is introduced into userspace, the name of the object shall also be included.

  The system-administrator must be able to selectively audit users.

- <u>Assurance</u>
  - <u>Operational Assurance</u>
    - <u>System Architecture</u>
      "The TCB shall maintain a domain for its own execution that protects it from external interference

or tampering(e.g., by modification of its code or data structures). Resources controlled by the TCB may be a defined subset of the subjects and objects in the ADP system. The TCB shall isolate the resources to be protected so that they are subject to the access control and auditing requirements."

The System Architecture should prevent anyone from modifying the TCB, which should be located in its own domain of execution. Resources should be protected so that they too are subject to all forms of access-control/auditing/accounting.

- System Integrity
  "Hardware and/or software features shall be provided that can be used to periodically validate the correct operation the on-site hardware and firmware elements of the TCB."

- Life-cycle Assurance
  - Security Testing
    "The security mechanisms of the ADP system shall be testedand found to work as claimed in the system documentation. Testing shall be done to assure that there are no obvious ways for an unauthorized user to bypass or otherwise defeat the security protection mechanisms of the TCB. Testing shall also include a search for obvious flaws that would allow violation of resource isolation, or that would permit unauthorized access to the audit or authentication data."

- Documentation
  - Security Features User's Guide
    "A single summary, chapter, or manual in user documentation shall describe the protection mechanisms provided by the TCB, guidelines on their use, and how they interact with one another.

  - Trusted Facility Manual
    "A manual addressed to the ADP system administrator shall present cautions about functions and privileges that should be controlled when running a secure facility. The procedures for examining and maintaining the audit files as well as the detailed audit record structure for each type of audit event shall be given."

  - Test Documentation
    "The system developer shall provide to the evaluators a document that describes the test plan, test procedures that show how the security mechanisms were tested, and results of the security mechanisms' functional testing."

  - Design Documentation
    "Documentation shall be available that provides a description of the manufacturer's philosophy of protection and an explanation of how this philosophy is translated into the TCB. If the TCB is composed of distinct modules, the interfaces between these modules shall be described."