Introduction to Realtimepublishers

By Sean Daily, Series Editor

The book you are about to enjoy represents an entirely new modality of publishing and a major first in the industry. The founding concept behind Realtimepublishers.com is the idea of providing readers with high-quality books about today’s most critical technology topics—at no cost to the reader. Although this feat might sound somewhat impossible to achieve, it is made possible through the vision and generosity of a corporate sponsor who agrees to bear the book’s production expenses and host the book on its Web site for the benefit of its Web site visitors.

It should be pointed out that the free nature of these publications does not in any way diminish their quality. Without reservation, I can tell you that the book that you’re now reading is the equivalent of any similar printed book you might find at your local bookstore—with the notable exception that it won’t cost you $30 to $80. The Realtimepublishers publishing model also provides other significant benefits. For example, the electronic nature of this book makes activities such as chapter updates and additions and the release of a new edition possible in a far shorter timeframe than is the case with conventional printed books. Because Realtimepublishers publishes our titles in “real-time”—that is, as chapters are written or revised by the author—you benefit from receiving the information immediately rather than having to wait months or years to receive a complete product.

Finally, I’d like to note that our books are by no means paid advertisements for the sponsor. Realtimepublishers is an independent publishing company and maintains, by written agreement with the sponsor, 100 percent editorial control over the content of our titles. It is my opinion that this system of content delivery not only is of immeasurable value to readers but also will hold a significant place in the future of publishing.

As the founder of Realtimepublishers, my raison d’être is to create “dream team” projects—that is, to locate and work only with the industry’s leading authors and sponsors, and publish books that help readers do their everyday jobs. To that end, I encourage and welcome your feedback on this or any other book in the Realtimepublishers.com series. If you would like to submit a comment, question, or suggestion, please do so by sending an email to feedback@realtimepublishers.com, leaving feedback on our Web site at http://www.realtimepublishers.com, or calling us at 800-509-0532 ext. 110.

Thanks for reading, and enjoy!

Sean Daily
Founder & CTO
Realtimepublishers.com, Inc.
# Table of Contents

Introduction to Realtimepublishers

Chapter 1: Windows Security Overview ............................................................... 1

Windows Security: A Broad Scope ................................................................. 1
  - Security Principals .................................................................................. 1
  - Network Security ................................................................................... 3
  - Physical Security .................................................................................... 4
  - Do the Right Thing ................................................................................ 5

Windows Security: The Forgotten Topics ....................................................... 5
  - Client Security ....................................................................................... 5
  - Built-In Vulnerabilities .......................................................................... 7
  - Active Directory .................................................................................... 8
  - File Servers .......................................................................................... 10
  - Servers and Services ............................................................................. 11
  - Software Management ........................................................................... 13
  - Network Security from the Inside Out .................................................. 18

Core Technological Issues ............................................................................. 19
  - Scripts as a Security Tool ..................................................................... 19
  - Shielding Windows Vulnerabilities ....................................................... 20
  - Defense in Depth .................................................................................. 22

Summary ........................................................................................................ 23

Chapter 2: Securing Clients ......................................................................... 24

Local Accounts and Passwords ...................................................................... 24

Service Management ..................................................................................... 28
  - Unnecessary Services .......................................................................... 28
  - Service Logons and Passwords ............................................................. 32

Local Firewalls ............................................................................................... 35

NTFS Permissions ......................................................................................... 37

Folder Redirection .......................................................................................... 41

Removable Storage ......................................................................................... 44

Local System Permissions .............................................................................. 46

Summary ........................................................................................................ 47

Chapter 3: Using Alternative Software to Reduce Your Attack Surface ....... 48

The Bundled Software .................................................................................... 48
# Table of Contents

What Will You Miss? .................................................................................................................... 49
  IE........................................................................................................................................... 50
  Outlook Express................................................................. 53
  Windows Media Player......................................................... 54
  Windows Messenger.......................................................... 55
Removing the Bundled Software .............................................................................................. 55
  IE........................................................................................................................................... 57
  Outlook Express................................................................. 59
  Windows Media Player......................................................... 61
  Windows Messenger.......................................................... 63
The Alternatives.......................................................................................................................... 65
  Web Browser ................................................................................................. 65
  Email Client ............................................................................................... 67
  Media Player ............................................................................................... 68
  Instant Messaging ....................................................................................... 69
Coexistence: The Middle Ground ............................................................................................ 70
Summary ....................................................................................................................................... 72
Chapter 4: Securing Active Directory .................................................................................. 73
Becoming Less Than Secure ..................................................................................................... 73
Getting AD Under Control......................................................................................................... 74
Using Granular Permissions ...................................................................................................... 78
  Permission Strategy ................................................................................................. 78
  The Delegation of Control Wizard ............................................................................ 79
  Manually Working with Permissions ........................................................................... 81
Maintaining Consistency ............................................................................................................ 84
  Command-Line Utilities ........................................................................................... 84
  Template-Based Administration ................................................................................ 85
Security Reporting ........................................................................................................................ 89
Overlooked (and Ignored) Security Problems ........................................................................ 92
  Unnecessary Security Principles ................................................................................ 92
  Disabled Accounts .................................................................................................. 93
  Accounts with Non-Expiring or Old Passwords ......................................................... 94
  Unlinked GPOs ....................................................................................................... 95
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary</td>
<td>96</td>
</tr>
<tr>
<td>Chapter 5: Securing File Servers</td>
<td>97</td>
</tr>
<tr>
<td>When Good File Servers Go Bad</td>
<td>97</td>
</tr>
<tr>
<td>What You Can’t See Won’t Hurt You</td>
<td>98</td>
</tr>
<tr>
<td>At the Network Level</td>
<td>100</td>
</tr>
<tr>
<td>Data Resilience</td>
<td>107</td>
</tr>
<tr>
<td>Volume Shadow Copy</td>
<td>107</td>
</tr>
<tr>
<td>Storage Management, Hierarchical Storage, and Antivirus</td>
<td>109</td>
</tr>
<tr>
<td>Applying Proper File Permissions</td>
<td>112</td>
</tr>
<tr>
<td>Maintaining Proper File Permissions</td>
<td>120</td>
</tr>
<tr>
<td>Summary</td>
<td>123</td>
</tr>
<tr>
<td>Chapter 6: Securing Servers and Services</td>
<td>124</td>
</tr>
<tr>
<td>Lock it Down</td>
<td>124</td>
</tr>
<tr>
<td>Security for Mere Mortals</td>
<td>125</td>
</tr>
<tr>
<td>How Big Is This Problem?</td>
<td>126</td>
</tr>
<tr>
<td>Running SCW</td>
<td>127</td>
</tr>
<tr>
<td>The Results</td>
<td>140</td>
</tr>
<tr>
<td>Beyond the SCW</td>
<td>141</td>
</tr>
<tr>
<td>Service Security</td>
<td>142</td>
</tr>
<tr>
<td>Access Controls</td>
<td>146</td>
</tr>
<tr>
<td>Finding Permissions</td>
<td>147</td>
</tr>
<tr>
<td>Permissions Reporting</td>
<td>150</td>
</tr>
<tr>
<td>Backing Up Permissions</td>
<td>151</td>
</tr>
<tr>
<td>Summary</td>
<td>152</td>
</tr>
<tr>
<td>Chapter 7: Security Through Software Maintenance and Filtering</td>
<td>153</td>
</tr>
<tr>
<td>What is Software Maintenance?</td>
<td>153</td>
</tr>
<tr>
<td>Patches, Patches, Patches</td>
<td>153</td>
</tr>
<tr>
<td>Must I patch?</td>
<td>154</td>
</tr>
<tr>
<td>Should I Test Patches First?</td>
<td>155</td>
</tr>
<tr>
<td>Windows Software Update Services</td>
<td>155</td>
</tr>
<tr>
<td>Microsoft Baseline Security Analyzer</td>
<td>157</td>
</tr>
<tr>
<td>Commercial Patch Management</td>
<td>160</td>
</tr>
<tr>
<td>Security Software Maintenance</td>
<td>164</td>
</tr>
<tr>
<td>Topic</td>
<td>Page</td>
</tr>
<tr>
<td>---------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>The Windows Security Center</td>
<td>164</td>
</tr>
<tr>
<td>Boundaries and the Multi-Layered Defense</td>
<td>166</td>
</tr>
<tr>
<td>What is Software Filtering?</td>
<td>167</td>
</tr>
<tr>
<td>Software Restriction Policies</td>
<td>168</td>
</tr>
<tr>
<td>Microsoft .NET Framework Security</td>
<td>175</td>
</tr>
<tr>
<td>Windows Script Host TrustPolicy</td>
<td>178</td>
</tr>
<tr>
<td>Summary</td>
<td>178</td>
</tr>
<tr>
<td>Chapter 8: Securing the Network</td>
<td>179</td>
</tr>
<tr>
<td>Security Through Architecture</td>
<td>179</td>
</tr>
<tr>
<td>Resource Clustering</td>
<td>179</td>
</tr>
<tr>
<td>Segmentation</td>
<td>182</td>
</tr>
<tr>
<td>Creating Trust Zones</td>
<td>183</td>
</tr>
<tr>
<td>The Architectural Big Picture</td>
<td>185</td>
</tr>
<tr>
<td>Security Through Monitoring</td>
<td>186</td>
</tr>
<tr>
<td>Quarantine</td>
<td>186</td>
</tr>
<tr>
<td>Network Sniffing</td>
<td>188</td>
</tr>
<tr>
<td>Port Monitoring</td>
<td>190</td>
</tr>
<tr>
<td>Intrusion Detection/Prevention Systems</td>
<td>193</td>
</tr>
<tr>
<td>Security Through Technology</td>
<td>198</td>
</tr>
<tr>
<td>802.1X</td>
<td>198</td>
</tr>
<tr>
<td>Wireless Security</td>
<td>199</td>
</tr>
<tr>
<td>Secure Network Adapters</td>
<td>200</td>
</tr>
<tr>
<td>IP Security</td>
<td>200</td>
</tr>
<tr>
<td>Summary</td>
<td>203</td>
</tr>
<tr>
<td>Content Central</td>
<td>203</td>
</tr>
<tr>
<td>Download Additional eBooks!</td>
<td>203</td>
</tr>
</tbody>
</table>
Copyright Statement

© 2005 Realtimepublishers.com, Inc. All rights reserved. This site contains materials that
have been created, developed, or commissioned by, and published with the permission
of, Realtimepublishers.com, Inc. (the “Materials”) and this site and any such Materials are
protected by international copyright and trademark laws.

THE MATERIALS ARE PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND,
EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE,
TITLE AND NON-INFRINGEMENT. The Materials are subject to change without notice
and do not represent a commitment on the part of Realtimepublishers.com, Inc or its web
site sponsors. In no event shall Realtimepublishers.com, Inc. or its web site sponsors be
held liable for technical or editorial errors or omissions contained in the Materials,
including without limitation, for any direct, indirect, incidental, special, exemplary or
consequential damages whatsoever resulting from the use of any information contained
in the Materials.

The Materials (including but not limited to the text, images, audio, and/or video) may not
be copied, reproduced, republished, uploaded, posted, transmitted, or distributed in any
way, in whole or in part, except that one copy may be downloaded for your personal, non-
commercial use on a single computer. In connection with such use, you may not modify
or obscure any copyright or other proprietary notice.

The Materials may contain trademarks, services marks and logos that are the property of
third parties. You are not permitted to use these trademarks, services marks or logos
without prior written consent of such third parties.

Realtimepublishers.com and the Realtimepublishers logo are registered in the US Patent
& Trademark Office. All other product or service names are the property of their
respective owners.

If you have any questions about these terms, or if you would like information about
licensing materials from Realtimepublishers.com, please contact us via e-mail at
info@realtimepublishers.com.
Chapter 1: Windows Security Overview

Microsoft Windows is the most popular desktop operating system (OS) in the world and holds a fair amount of the server OS market share as well. Love it or hate it, Windows has a place in nearly every enterprise, and like any OS, Windows has unique security problems and strengths. Today’s enterprise is placing a greater focus on security—an endeavor driven in no small part by the array of new laws and regulations with a security focus—and securing Windows is becoming an increasingly important task.

You can spend a lifetime attempting to master Windows security and spend much of your free time discussing and debating the topic with other IT professionals. However, this guide will concentrate on security as more than just philosophy and policies—security is a practical topic with real-world impact, which is the focus of this guide.

Windows Security: A Broad Scope

So much has been written about Windows security that it is tempting for me to simply refer you to the bevy of existing content and tell you to “do whatever they say.” The fact is that you’re probably sick and tired of being told to lock down your firewall, get those patches deployed, and keep the antivirus software updated—security standards that are obviously valuable but are well covered in most texts. That said, there are some security basics that form the foundation for a secure enterprise, and although you’ve hopefully had them in place for years, they’re always worth a quick review.

Security Principals

Managing security principals—user and computer accounts—is obviously at the heart of any security plan. Most organizations still rely on passwords and user names for identity management. Tools such as RainbowCrack, which can provide clear-text passwords for even complex user accounts in just a few minutes, reinforce the need for effective password management policies as a fundamental good practice (require frequent password changes, accustom users to employing long passphrases rather than passwords, and so forth). Ideally, switching to less easily compromised identity management techniques—biometrics, one-time tokens, smart cards, and so forth—can provide a strong level of security and eliminates the burden on users to choose (and remember) complex passphrases.
RainbowCrack: Passwords on Demand

Fully appreciating the need for frequent password changes and complex passphrases requires an understanding of how commonly used tools such as RainbowCrack work. Windows doesn’t store passwords. Instead, it encrypts passwords using a one-way hash function, then stores the result. The purpose of the hash is to generate an encrypted string that cannot then be unencrypted: In fact, the hash produces only part of the final encrypted product, meaning Windows is only storing part of an encrypted string. Without the other part—which is discarded—you can’t retrieve the original, unencrypted password.

The following math problem illustrates this concept: 5 divided by 2 is 2 with a remainder of 1 (2R1). If you were given any two of those components (5, 2, or the answer of 2R1), you could solve for the answer. For example, x divided by 2 is 2R1; x is obviously 5. In a one-way hash, however, the remainder is thrown away. If you’re given the result, x divided by 2 is 2, you can’t solve for the correct result of x=5.

Windows goes a step further by never transmitting the hashed password on the network. Instead, clients use the hashed password as an encryption key to encrypt an authentication packet. A domain controller then uses its own stored copy of the hash as the decryption key—if it can decrypt the packet, the password must have been typed correctly.

The trick to RainbowCrack is that the domain controller stores the hashed copies. The other trick is that the hash algorithm used by Windows is well-known and documented. It’s a fact of encryption that if you take any given clear-text data and encrypt it using a specific algorithm, you will always get the same encrypted result.

RainbowCrack generates millions of clear-text passwords, then hashes them using the same algorithm that Windows uses to hash passwords. The result is a table of clear-text passwords and their corresponding hashes. If an attacker gains access to a domain controller’s hashed passwords, the attacker can simply look for matches in RainbowCrack’s tables and discover the clear-text password.

The catch is that RainbowCrack can take a long time to generate a table with passwords that are sufficiently complex. A DVD set, comprising more than 100GB of data, is available on the Internet that contains RainbowCrack tables for seven-character passwords containing any combination of characters. By simply plunking down a couple of hundred bucks for these DVDs, an attacker can instantly discover any seven-character password. Thus, longer passphrases—longer than 10 characters, if possible—are considered more secure. The computing power and time required for a tool such as RainbowCrack to generate tables for a 10-character or longer passphrase is tremendous, and the likelihood that someone will spend this amount of time is small.

Often-overlooked local accounts pose the greatest password threat. The local hashed passwords are typically much easier to access than a domain controller’s passwords, and by gaining access to a local Administrator account on a workstation or server, an attacker can do plenty of damage, such as installing keystroke logging software that will record users’ activity.

Most organizations have a fairly good enterprise password policy in place. Windows Server 2003 (WS2K3) domains, for example, default to requiring a fairly long, strong password—one composed of upper and lowercase letters as well as numbers or symbols—and default to a reasonably aggressive password age policy. These password policies are often implemented across systems in enterprises so that vertical applications, other OSs, and so forth have strong password policies requiring relatively frequent password changes. But password best practices seem to stop at the enterprise level, never finding their way down to local computers’ user accounts and other often-overlooked areas. This guide will focus on better management of these equally critical yet often-overlooked security principals.
**Network Security**

As with password policies, enterprises have a pretty good grasp of effective network security practices, starting with firewalls, firewalls, firewalls. In fact, some enterprises even segregate their internal networks with firewalls; doing so limits cross-segment access and helps to contain attacks or viruses should they occur. Figure 1.1 shows an example of this segregation technique, where access to shared resources on servers is relatively unrestricted and uniform, but cross-segment access is severely restricted or even disallowed altogether.

![Figure 1.1: Using firewalls to compartmentalize a network.](image)

The theory behind internal segregation is that network segments containing client computers are less likely to have shared resources; organizations using this technique don’t use client-based file and print sharing and instead consolidate shared resources on servers. As a result, clients have no need to connect to one another, and segregating them with firewalls helps prevent (or at least hinder) the spread of viruses and other malware.
Of course, there is much more to network security than just an effective firewall. Some organizations use technologies such as IP Security (IPSec) to encrypt and digitally sign important traffic, preventing both data spoofing and electronic eavesdropping of confidential data. Some companies implement low-level security such as 802.1X to prevent unauthorized computers from even transporting data across wired or wireless networks. Network adapter manufacturers are beginning to ship adapters with embedded digital certificates, allowing the adapter itself to authenticate to the network and gain access, preventing unauthenticated cards from acquiring a TCP/IP address.

Many organizations are so concerned about physical network security—such as the possibility of someone plugging a computer into a lobby network jack and eavesdropping on the corporate network—that they invest in special network adapters that automatically encrypt everything coming in and out by using hardware-level IPSec. This technique uses virtually no computing resources in the computer because the adapter has IPSec-specific hardware onboard and, once deployed, is relatively easy to manage and maintain.

Network security also encompasses OS security procedures such as uninstalling unnecessary software and services to remove potential vulnerabilities, keeping software fully patched and updated to defeat vulnerability exploits, and so forth. As I’ve mentioned, security is a broad, pervasive topic that affects every aspect of enterprise IT life. There are plenty of important network security practices and procedures that are routinely overlooked, which we will explore throughout this guide.

**Physical Security**

Physical security is important—leaving the data center unlocked leaves your servers as open to attack as leaving the firewall open; allowing someone to walk away with a server’s removable hard drive defeats the purpose of file-and-folder security, password management, and network security. Thus, enterprises spend a lot of money on physical security measures such as electronic card-key lock systems and restricted data center access.

However, the physical security of *data* is often overlooked. Companies who spend thousands of man-hours applying file and folder permissions will allow users to print those secured files and leave the hardcopy lying around for anyone to see. These bypasses of electronic security are a major weakness in many organizations; some organizations fight back with aggressive user education, easy access to locked filing cabinets and document shredders, and so forth, trying to do a better job of matching the physical security of their data to the electronic security measures.

Some enterprising administrators will refuse to implement any form of electronic security that can be readily bypassed through a lack of physical security. In fact, businesses would be better off writing policies that address security needs in general—statements such as “documents containing customer information must not be accessible to individuals outside the company”—then requiring each portion of the company to comply. Network administrators would implement appropriate electronic security measures while facilities managers could implement the appropriate locked filing cabinets, paper shredders, and so forth; users would be required by this technology-agnostic policy to use both electronic and physical security measures.
Do the Right Thing

In the end, most enterprises want to do the right thing with security. Inevitably, however, things are overlooked, often because Windows can make it terribly difficult to do the right thing. For example, what if you want to change the password for the local Administrator account on every Windows 2000 (Win2K) or Windows XP client computer on a regular basis? Talk about painful: Windows provides absolutely no tools for making this task feasible; thus, most organizations simply ignore it. The result: The Forgotten Topics of Windows Security—considerations that are overlooked because dealing with them is impossible from a practical point of view.

Windows Security: The Forgotten Topics

Given that you’ve probably heard chapter and verse on password policies, firewalls, viruses, patches, and the other “big” topics of enterprise security, this guide will focus on addressing the forgotten topics—security issues that are overlooked because they are too easy to overlook, too difficult to deal with, or simply so well hidden within Windows that you don’t realize they exist. Popular topics such as firewalls and smart cards are well-known and well-understood, and if your organization isn’t where it should be on those topics, it is probably because of major business concerns (implementing smart cards, for example, isn’t cheap). But the forgotten topics of Windows security can be addressed, if you know they exist, and if you have the right tools on hand.

Client Security

Client computers are probably the least secure portion of any enterprise simply because there are so many of them and they’re so difficult to completely secure. Laptops come and go, following employees from work to home and on the road, and they’re difficult to lock down while still making them usable in their various roles. Windows client computers have as many security concerns as Windows servers—after all, they’re pretty much the exact same OS—but servers, with their centralized locations and more rigid roles, are given much more attention when it comes to security.

Returning to the example of local Administrator account passwords, client computers often store sensitive data, but their Administrator accounts are often so rarely maintained that the client computer represents a more likely target for an attacker than the server on which that sensitive data started. NTFS permissions on clients are rarely as well thought-out as the permissions on servers, but clients store just as much sensitive data in most organizations. Clients typically run a host of services that aren’t needed, and each of them represents a potential security vulnerability, increasing your software maintenance overhead should an exploit be discovered.

Managing local computer groups—especially the Administrators group—is as important as managing client computers. Windows provides some help in this regard, providing security template capabilities that allow you to control membership of these groups through a centralized Group Policy Object (GPO), for example. However, because so many administrators don’t realize the dangerous capabilities of this group or don’t realize how relatively easy it is to control group membership centrally, the problem is overlooked as a sort of “nothing can be done about it” issue.
Client computers usually come equipped with removable storage—optical media burners, Universal Serial Bus (USB) flash drives, and so forth—that enable users to bypass carefully constructed file server security schemes and walk out of the office with confidential data tucked away in a purse, pocket, or briefcase. In the past, companies often purchased tools that would disable floppy drives or purchased computers without floppy drives installed; while companies can easily purchase client computers that lack optical burners, it’s impossible to buy a computer without USB ports! So many keyboards, mice, and other peripherals rely on USB that not having them isn’t a practical option; so what can be done about removable USB storage?

There are ways around some of the problems. For example, local user account management is a challenge you can conquer with the right tools, which we’ll discuss in detail in Chapter 2. Dealing with client file security is something you can often dispense of entirely by using clever folder redirection schemes. For example, as Figure 1.2 shows, users can access local folders that are actually transparently redirecting to a more easily secured file server.

Figure 1.2: Redirecting local folders provides users with convenience while actually keeping files in a more readily secured location.
Windows XP Service Pack 2 (SP2)—Microsoft’s latest security volley—provides an opportunity for enterprises to shield many of Windows XP’s vulnerabilities—including future, undiscovered ones—by strictly controlling incoming traffic. Of course, the firewall can also complicate remote client management, which means you’ll need to make some careful design and implementation decisions regarding this important new security tool.

Chapter 2 will explore the Windows Firewall, explain how it works and what it does, and offer some advice and tools for managing it more effectively.

Built-In Vulnerabilities
Microsoft certainly doesn’t build vulnerabilities into Windows intentionally, but some components of Windows certainly are troublesome from a security point of view. Internet Explorer (IE) is perhaps the most well-known culprit, but Windows is a very, very large OS with millions of lines of code—vulnerabilities exist everywhere. Removing or disabling these vulnerabilities and replacing their functionality with other products can provide a more secure system.

Chapter 3 will discuss software alternatives for various Windows components, including information about how you can implement these alternatives and remove or disable their built-in counterparts.

Several utilities available on the Internet purport to remove IE. In fact, doing so is very difficult. Although the IE application (iexplore.exe) can be removed, the guts of IE—its HTML rendering engine and other internal components—are part of the Microsoft Foundation Classes (MFC), and removing those would break Windows entirely. In Chapter 3, we’ll explore how to entirely disable IE.

IE is an often-targeted Windows component, and its tight integration with other Windows components often allows IE vulnerabilities to deeply affect the OS; Figure 1.3 shows the Computer Emergency Response Team (CERT) vulnerability list for IE, which, as of January 2005, included nearly 30 advisories. Mozilla’s Firefox at the time had no listings—a marked contrast, and a strong argument for considering a switch.
Active Directory

Active Directory (AD)—central as it is to Microsoft’s enterprise security strategies—receives plenty of attention from enterprise administrators, yet still has several important security considerations that are often overlooked. For example, many organizations leave AD’s permissions at their default settings, using all-powerful Domain Administrator accounts to perform management. In fact, AD can be made more secure by applying more granular, customized permissions that allow less-privileged accounts to perform day-to-day administration tasks, reserving the Domain Administrator accounts for less frequently performed tasks. Of course, setting up these more complex permissions can be difficult to manage in the long-term, so many organizations just forget about it. Certainly, AD’s security management isn’t exactly intuitive—by default, organizational unit (OU) security isn’t even visible in the Active Directory Users and Computers console. As Figure 1.4 shows, enabling the security features merely provides access to a bewildering interface.
Chapter 1

Figure 1.4: Reviewing permissions in AD Windows’ built-in user interface.

Also, many organizations’ AD installations start off great: well-planned, well-secured, and well-implemented. As the business changes and grows, however, the AD configuration tends to evolve into a less secure state. This decline isn’t anyone’s fault, it is just the natural result of working with such a complex product using the relatively unsophisticated tools that are built-in to Windows.

Part of the problem with AD’s security is that it’s so flexible, but provides so little documentation and change control built-in. Over time, administrators tend to apply patch-fixes to problems—giving one user a certain permission on a certain directory object, denying another group permission to a specific attribute, and so forth. These spot-fixes make the directory permissions more difficult to manage, making administrators yearn for a tool that will help them determine what security is in place as well as ensure that the current settings are as efficient as possible.

The “last guy” syndrome is especially common in AD: You wouldn’t apply that sort of spot-fix, but you’re new to the company; the “last guy” used many individual permissions and now they’re scattered all over the directory—and you’re afraid to change anything for fear of what will break.

These concerns and additional issues will be addressed in Chapter 4.
Chapter 1

File Servers

File servers present a major set of challenges to enterprise security. Windows file permissions and file permission inheritance is a powerful, flexible tool for securing files, but they can also be complex, and the built-in user interface for managing file permissions doesn’t necessarily reflect that complexity. Thus, organizations tend to manage file server permissions somewhat haphazardly, rarely enjoying any kind of insight into their total security picture. It becomes easy to have files that are incorrectly secured, simply through oversight.

Windows’ own file-management routines also make security management more difficult than it needs to be. For example, most files take their security permissions from their parent folders rather than having those permissions directly applied. This technique makes security management somewhat less complex because you can manage security on fewer objects (you’re likely to have fewer folders than files). When you move or copy a file, however, its permissions inherit from its new parent folder. Although this feature is useful in some instances because you can simply relocate files to change their permissions (assuming all of your folder permissions are correct), it is an annoying feature because you can’t easily relocate files en masse (such as to a new volume) while retaining their permissions. Of course, there are third-party tools and Microsoft resource kit tools that can assist with this task, which we’ll explore in Chapter 5.

Tradtitionally, Microsoft’s resource kit tools are created by groups within Microsoft who recognize a need for them. In fact, many aspects of Windows management would be horrible without these tools. However, Microsoft doesn’t support these tools, so the very thing you need to get your job done is, in fact, unsupported.

When available and affordable, it is preferable to work with commercial tools from third-party companies because they are supported. Thus, if there is a problem with the way a tool works, you have somewhere to turn. In addition, third-party tools tend to be more robust and are better documented.

Reporting and auditing is another area in which Windows falls short. Many organizations—especially those affected by legislation mandating minimum security standards—often have a need to generate a comprehensive report about their security settings. For example, you might want to create a report of all files to which a certain user group has access, all security settings on a set of files, or files that don’t have a specified set of permissions applied. These tasks are impossible with Windows’ built-in tools, but certainly need to be addressed in a timely manner.

Another security shortcoming in Windows that most administrators don’t even realize exists is that, by default, users accessing a Windows file server can see every folder and file—even ones they don’t necessarily have access to. Preventing users from seeing files they shouldn’t have access to is the first step in keeping those same users from trying to work around your security restrictions. Although simply hiding files doesn’t protect them, obscuring files that are already protected can make those protections more effective by helping to keep users from trying to work around the protections.

Chapter 5 will look at ways to help keep users focused and on-task by hiding resources to which they don’t have access.
Servers and Services

The background services that run on various servers present a double security problem. On one hand, each service naturally has some kind of built-in functionality and most are designed to be network-accessible. Thus, services can have vulnerabilities in their functionality, which can be exploited over the network. On the other hand, most services must authenticate in order to run. Although many use the all-powerful LocalSystem account (a problem in and of itself), others use a domain or local user account, which must be managed. Figure 1.5 illustrates service authentication and network access.

Figure 1.5: Services present a dual security issue through access and authentication.
Several parts of service management are frequently overlooked:

- Unnecessary services are often allowed to remain installed and running, presenting an unnecessary vulnerability on the server.
- Services are often left running under all-powerful accounts (like a Domain Admin account or the LocalSystem account) when doing so isn’t necessary, thereby giving services—and anyone who manages to exploit a vulnerability in the service—excess privilege.
- Service accounts—when LocalSystem isn’t used—are often allowed to run for months or years without a password change, increasing the risk that a password will be compromised.

As with the other security problems discussed so far, these result primarily from Windows’ own lack of ability to easily manage services.

![Chapter 6 will explore tools and techniques you can use to correct this often-overlooked area of security, eliminating unnecessary services and bringing better management to service accounts.](image)

Don’t think that disabling an unnecessary service will protect you. Doing so will make the service less accessible, certainly, but anything that can be disabled can also be re-enabled, restoring the service—and any vulnerabilities it may contain—to service. The best step is to uninstall the software that installed the service in the first place, thus removing the binary code from the computer altogether.

However, this method is not always an option, particularly for services that are installed with Windows itself. In those instances, deleting the service definition will make it much more difficult to restore the service to operation. You can also implement a Software Restriction Policy (SRP—available on Windows XP and later) that prevents the service’s executable from being run by the OS.

Don’t think that you can delete a service’s executable: Most built-in services are under Windows File Protection and will be restored automatically at some point. At the very least, they will be copied back by the next service pack you install.

![Chapter 6 will also look at several often-overlooked security problems such as the management of registry permissions, management of software allowed to run on servers, and so forth.](image)

Because servers typically represent such a valuable asset in an enterprise—storing most of an organization’s critical data resources—servers merit much tighter management than clients. With the right tools and techniques, it is possible to manage to an effectively detailed level.
Software Management

Software management covers a few often-overlooked areas of security. For example, despite the nearly constant coverage by industry media, organizations with Windows environments still tend to fall behind on patch management. It is easy to understand why: Most patches don’t fix bugs that crop up very often, so applying a patch seems suspiciously like fixing something that isn’t broken. Patches that fix security vulnerabilities are often given short shrift too, with many organizations adopting a “We’ve never been hit, so why bother?” attitude. Compounded this attitude with the fact that Windows patch management can be extremely difficult to manage, and patch management winds up being another overlooked security problem.

To improve the situation, Microsoft released Software Update Services (and later renamed it Windows Update Services), which pulls software updates from Microsoft’s servers, allows a local administrator to test and approve them, then deploys those updates to all Windows computers (newer ones, at least), as Figure 1.6 shows. Software Update Services (and Windows Update Services) is a great free tool, but it focuses entirely on updates for the Windows OS; it doesn’t cover applications (Windows Update Services does cover Office applications and most Microsoft server products, but not third-party applications). Software Update Services isn’t intended as a replacement to more robust software deployment or patch management software; it’s intended to provide the minimal functionality a Windows enterprise needs to start getting a handle on patch management.
Of course, there are third-party tools and some clever techniques that can make patch management easier and more effective, which we’ll explore in Chapter 7.

There is more to software management than software maintenance. As Chapter 7 discusses, software filtering is an oft-overlooked security consideration. The idea is to compile a complete list of the software required in your organization, then configure Windows (or a third-party tool) to allow only that software to run. Viruses, spyware, and other malware quickly become a thing of the past as any software you don’t specifically allow is automatically denied. It can take a lot of work to set up, but once in place, you’ll have a more secure environment that is based on positive identification of allowed software.
Versions of Windows back to Windows 95 had the capability to prevent specific software from running, but those capabilities were awkward to use and not really capable of handling large lists of software; they were intended to keep users from running games and such. Plus, you had to specify the software that wasn’t allowed; nowadays, that is a big list.

Microsoft’s answer is Software Restriction Policies, which is available on Windows XP and later versions of Windows (including WS2K3). Third-party tools can also help prevent unwanted software from running and can often provide more granular management and configuration, as Chapter 7 explores.

While on the subject of software that shouldn’t be allowed to run in your environment, let’s not forget about viruses and other forms of malware. It’s quite a diverse software category, encompassing:

- **Viruses**—These applications are designed to do damage, either to an individual computer or to multiple computers. They’re also designed, like biological viruses, to self-replicate, spreading themselves by way of a variety of means to multiple computers.

- **Spyware**—These applications typically watch specific user actions, logging them and reporting them back to their creators. The least-malicious spyware might log users’ Web-surfing habits for demographic purposes; more devious software might log users’ keystrokes, allowing the spyware’s creators to more easily bypass your corporate defenses.

- **Adware**—These applications work by popping up advertising, often inside your users’ Web browsers (but also outside of it). They disrupt productivity, waste computing resources, and are frequently packaged with spyware of some kind.

Viruses can be creative in the ways they damage your network. For example, if someone wants to attack your internal mail server, doing so from outside your network can be difficult. Internal servers often aren’t exposed on the Internet (they might, for example, use a gateway to relay external email) and don’t often access the Internet. They don’t run easily infected Web browsers (that is, they often aren’t used for Web browsing at all), so they’re difficult targets. However, as Figure 1.7 shows, your client computers can be more easily infected with malware from a Web server that they visit. Once a single computer is infected, your entire environment is open to infection.
The attacking virus might have no effect on the client computers at all. Instead, it waits until its numbers have grown and launches a denial of service (DoS) or similar attack on your now-easily-accessible mail server, as Figure 1.8 illustrates.
These attacks can be difficult to defend against once they begin, meaning you must stop them from beginning. Doing so requires not only antivirus software on your clients but also up-to-date antivirus software and anti-spyware software capable of stopping other categories of malware. Viral attacks of this type are often overlooked: Administrators tend to assume that a virus intends to damage the machine it infects. This assumption leads administrators to leave low-risk machines—ones not containing critical data—unprotected from viruses. The machines are therefore vulnerable and put at risk the rest of the network as the virus that infects the supposedly low-risk machines might have bigger plans than damaging a single low-risk machine.
Chapter 1

Network Security from the Inside Out

Overall network security is often overlooked in many Windows environments. In this context, overall security does not refer to basics such as firewalls; those are pretty well-understood and widely used. The problem is that firewalls only protect you from threats originating outside your network, and plenty of attacks—most of them, in some folks’ view—occur right inside your network, behind the firewall. Tools such as Intrusion Detection Systems (IDSs) and Intrusion Prevention Systems (IPSs) continually monitor network traffic to look for signs of attack (DoS attacks, certain types of traffic associated with active viruses, and so forth). An IDS will call your attention to these signs of attack; an IPS can take further steps to stop the attack or further protect the target of the attack.

There are other technologies designed to help you better secure your internal network. IPSec and 802.1X can help ensure that only authorized computers can connect to and communicate on your network, and can help encrypt traffic so that electronic eavesdropping is nearly impossible. Although much of the functionality needed to support these technologies is built-in to Windows (and has been since Win2K), not many organizations make use of them because these technologies are perceived as difficult to configure and maintain.

Chapter 8 will look at these and other technologies designed to help secure your network infrastructure.

Leave No Path Unsecured

Securing the network is more than just good sense, it’s absolutely essential to creating a more secure Windows environment. Every aspect of your enterprise must have a roughly equal level of security or something becomes a weak link.

Microsoft learned this lesson when creating Digital Rights Management (DRM) software for Windows Media audio files. Initially, Microsoft’s DRM efforts focused on encrypting files and so forth, with the intent to protect the media. But the company left a weak link—the audio output. Attackers wrote custom audio drivers that intercept the audio output after the DRM software had decrypted it, then save the now-unprotected audio content to a separate file. Microsoft corrected the problem in Windows XP by implementing the Secure Audio Path (SAP). Now, Windows Media Player won’t play protected content unless every audio driver involved in the playback is digitally signed by Microsoft.

In this situation, SAP is equivalent to your network infrastructure. Securing data with strong passwords and encryption is useless if the path that data travels—your network—is insecure. By better securing the infrastructure, creating, if you will, a “secure data path,” you can create a more secure Windows enterprise.
Core Technological Issues

When it comes time to better secure your environment, there are a few core technologies you’ll need to become familiar with. Like all technologies, these have some drawbacks and disadvantages that you should be aware of so that you can make informed decisions about the technologies that are used in your environment.

Scripts as a Security Tool

In the past year or so, scripting has become more popular for Windows administrators. Once considered primarily a tool for writing logon scripts, scripts are now considered by many administrators as tools to perform any number of administrative tasks.

Scripts can do some great work. For example, a script can be used to quickly change the password that a service, running on a remote computer, uses to log in. Such scripts are typically short and fairly easy to write.

```vbscript
strComputer = "Server2"
Set objWMIService = GetObject("winmgmts:" & "\" & _
 strivingLevel=impersonate)!" & _
 strComputer & "\root\cimv2")

Set colServiceList = objWMIService.ExecQuery _
    ("Select * from Win32_Service Where StartName = '.\netsvc'")

For Each objService in colServiceList
    errReturn = objService.Change( , , , , , , "password")
Next
```

So why pay for the third-party tool when you’ve got a script that will do it for free? The reason is that scripts can’t generally duplicate the functionality of a well-written application designed for the same purpose.

In addition, scripts take a special skill set to write. Although more administrators are working to gain this skill set, few organizations have more than one or two administrators—the “scripting guys”—who understand scripting in sufficient detail to use it effectively. Thus, any scripts those administrators write will be difficult to use or maintain once those administrators are gone, creating a significant business continuity issue. For example, if your organization becomes accustomed to fulfilling legal obligations—such as reporting on file security settings—by using a script, once the person who wrote that script leaves the company, you’re in a difficult position.
Administrators new to scripting often bring a different type of problem. Because scripts are so easy to write (physically, that is; you just need Windows Notepad) and execute, newer scripters tend to fire off scripts in the production environment without adequately testing them first. The consequences are obvious.

Finally, while scripts can be a useful administrative aid, they’re also the preferred vehicle for many types of malware. Email-based scripts, for example, have accounted for the last several major viruses to strike enterprises. Many organizations have a reasonable reaction to this situation, which is to take measures to prevent scripts of any kind from running. Scripts can be secured, particularly using the Windows Script Host’s TrustPolicy settings, but doing so takes additional effort and expense.

The bottom line? Scripting can be a valuable administrative tool, but its use should be restricted to organizations that want to make it a formal, supported part of their environments. Script authors should be trained to thoroughly document their scripts, practice source code control and change control, test their scripts in a test environment, and follow organizational guidelines for software deployment when using their scripts in production. The scripting environment must be secured to prevent malicious scripts from running while allowing approved scripts to execute. Many organizations have too many projects on their plates already, making it difficult to incorporate scripting as a formal type of project. Those organizations should stick with existing tools from Microsoft and third-party vendors.

**Shielding Windows Vulnerabilities**

Windows comes from Microsoft fully loaded with a host of security vulnerabilities. These vulnerabilities can actually be broken down into two classes:

- Vulnerabilities that are the result of the way a Windows function or feature operates, such as the way IE makes it relatively easy for spyware to install itself on users’ computers.
- Vulnerabilities that are unintended and the result of a bug.

The problem with both of these categories is that they’re difficult to predict, locate, fix, and remove without Microsoft’s help. It took Windows XP SP2, for example, to provide a modicum of relief to IE’s wide-open policy on installing potentially unwanted software. Bugs, of course, get fixed when they’re discovered and Microsoft issues a patch. The lag time between bug discovery and patch availability can be significant. In the recent example of the JPEG rendering vulnerability in Microsoft’s Graphical Device Interface (GDI), the patch took several weeks to become available.

In the meantime, all you can do is try to prevent these vulnerabilities from being reached. One way to do so is a local firewall, such as the Windows Firewall (included with Windows XP SP2) or any of a number of third-party firewall software packages. As Figure 1.9 shows, these tools don’t fix the vulnerabilities within Windows; they simply prevent malicious software from taking advantage of the vulnerabilities by severely restricting access to the computer from the network.
Local firewalls can help address a variety of security problems, such as allowing unnecessary services to continue running. By blocking access to these services using a firewall, you don’t need to worry about the service being exploited.

Local firewalls are typically stateful firewalls. In other words, they allow all outgoing traffic and allow incoming replies to outgoing traffic automatically. They block all incoming traffic unless you create an exception that allows such traffic. Figure 1.10 illustrates this technique, which is designed to allow maximum local functionality while minimizing management overhead. For example, you don’t need to explicitly allow incoming HTTP traffic, because the stateful inspection process will ensure that HTTP traffic that is a reply—such as incoming traffic from a Web server that the user is accessing—will automatically be allowed.
**Defense in Depth**

One key element in any security plan is to assume that every defense you have will fail, and to put backups in place to handle that situation. For example, where should you install virus scanners? The answer: Everywhere, as shown by the magnifying glasses in Figure 1.11. Install them at your mail gateway, your Internet gateway, on client computers, on server computers, on proxy servers, and anyplace else you can install an antivirus scanner. Doing so will improve the odds that every virus will be caught.

> Ideally, use scanners from different vendors. Each antivirus vendor has different strengths and weaknesses; by using a variety, you’ll take advantage of all their strengths while allowing them each to cover for the others’ weaknesses.

![Figure 1.11: Install virus scanners everywhere on your network.](image)

Defense-in-depth applies to every security measure: If one firewall is good, several firewalls might be better. For example, protect each category of Internet-accessible resources with a firewall, as Figure 1.12 illustrates. Separate public Web servers from the Internet with a firewall and from your intranet with another. Ditto for extranet servers, which deserve their own sets of firewalls. Successive firewall layers should be from different vendors. Although this setup might increase management overhead somewhat for the initial deployment, it means that weaknesses of one firewall product can’t be exploited to create an open channel into your intranet.
And always remember: The most secure connection is the one that doesn’t exist. If intranet clients have no need to access your extranet, for example, then don’t create a connection between the two networks (if the connection is needed, then use a firewall to protect it, of course).

**Summary**

There’s no doubt that Windows security is a broad topic with many areas for concern. However, many security issues are often overlooked, simply because they’re difficult to implement, manage, discover, or understand. This chapter has introduced you to a few often-overlooked security areas; throughout the rest of this guide, we’ll explore these overlooked areas in more detail and discuss practical advice for handling them. An introduction to tools and techniques that can overcome Windows’ shortcomings and missing capabilities will help you to develop a more comprehensive, detailed, and functional security plan for any Windows enterprise.
Chapter 2: Securing Clients

Client computers are often neglected when it comes to security. Everyone tends to focus on servers, and, let’s face it, servers are definitely easier to secure. In addition, servers exist in data centers or other protected locations and are tended to by trained administrators. Client computers, in contrast, sit on the desks and in the carrying bags of mere users, and are subjected to every imaginable stress: physical security threats, spyware, viruses, airports, hotels, and so on. The reality is that client computers can hold just as much critical information as servers. On their client computers, users store local copies of files (the only copy of those files, in some cases), use Windows’ Offline Files features to retain copies of server-based data, and so forth. The amount of corporate information stored in users’ mailboxes, for example, is staggering—as much as 70 percent, according to a recent survey by VERITAS. Corporate confidential data is more likely to be compromised from a client computer than from a server, yet client computers typically have the least amount of security and the poorest, from a security viewpoint, configurations. This chapter will highlight some of the major security concerns affecting client computers, and give you ideas about how to address them.

Local Accounts and Passwords

Local user accounts exist on every Windows computer except domain controllers, and client computers’ local accounts are one of the most-overlooked security issues in any enterprise. For example, every computer has a local Administrator account, which has complete and total control over nearly everything on that computer—including the profile contents for domain users who utilize the computer. Simply gaining access to the local Administrator account can therefore provide access to a great deal of domain information, even though the local Administrator account doesn’t have direct access to anything in the domain.

If the local Administrator account is compromised, a keystroke logger could be installed, enabling the hacker to compromise credentials of other users that may have access to sensitive data on the servers. There are software utilities, such as PestPatrol, available that can scan for keystroke logging tools.

Some organizations will take the time to configure local account policies, governing the maximum age, minimum length, and other restrictions for the local accounts. This process is simple, as Windows Group Policy allows you to do so through Active Directory (AD). Figure 2.1 shows an open Group Policy Object (GPO) that is linked to an organizational unit (OU); every computer within this OU will be affected by the password policy configured in the GPO.
Figure 2.1: Configuring password policy in a GPO.

However, local accounts aren’t used all that often in many environments. If an account—such as Administrator—isn’t used, then its password will never be changed and it remains a security liability. Perhaps the most common way of dealing with this local account security issue is to simply ignore it, creating one of the biggest issues in client computer security. Many organizations take half the time on local account management than they spend managing their domain accounts, yet those local accounts can have access to just as much sensitive data.

Local computer accounts don’t have direct rights to resources stored in a domain. But when domain information is copied to a local computer, local accounts—especially the Administrator account—can gain access to it, essentially bypassing domain security (from a business viewpoint, at least) now that the file is under the computer’s local security.
One way to quickly deal with the issue is to write a short script. The following VBScript, for example, can be used to change the local Administrator password on a remote computer:

```vbscript
sComputer = "client1"
Set oUser = GetObject("WinNT://" & sComputer & "/Administrator, user")
oUser.SetPassword "N3wP@ssw0rd!"
oUser.SetInfo
```

Naturally, this isn’t a terribly useful tool because it only changes one computer at a time. A more powerful version of this script would read all of the computers from a file, listing one computer per line, and change their local Administrator accounts:

```vbscript
Set oFSO = CreateObject("Scripting.FileSystemObject")
Set oTS = oFSO.OpenTextFile("C:\Computers.txt")
Do Until oTS.AtEndOfStream
    sComputer = oTS.ReadLine
    Set oUser = GetObject("WinNT://" & sComputer & "/Administrator, user")
oUser.SetPassword "N3wP@ssw0rd!"
oUser.SetInfo
Loop
oTS.Close
```

However, this solution is still not ideal. It requires that you maintain a huge list of client names—a task that makes it easy to miss one. In addition, any computer that isn’t available (turned on) when you run the script won’t be updated—in fact, this script will crash on the first unavailable computer. It’s possible to change the script so that it will log unreachable computers, and you can even have it read computer names from AD. However, even this solution doesn’t address all the shortcomings. AD too often contains old computer accounts, and might not contain the name of every computer in your environment (standalone lab computers, for example). Smart organizations will rename the local Administrator account, but might not have done so consistently on every computer. In this situation, you need to change the password of an account whose name you don’t even know—a difficult task!

Commercial tools can do a better job in many cases. For example, Absolute Dynamics’ cPWD can change passwords on multiple computers, and even target computers on which the Administrator account has been renamed. As Figure 2.2 shows, multiple computers have been targeted to have their local Administrator password changed.
Figure 2.2: cPWD makes changing local accounts easier.

Simply entering *A* for the account name will target the local Administrator account by its security identifier (SID), regardless of the account’s actual name. The tool is designed to target a list of computers, but has the capability to dynamically generate that list, as Figure 2.3 shows.

Figure 2.3: Dynamically targeting computers to change local account passwords.

This scan is performed through the browse master, meaning it will pick up only those computers that are online at the time. Computers that are offline will be missed, and you’ll need to pick up those separately—perhaps by running the scan several times each month to get as many computers as possible.

Like so many security issues that affect client computers, local account passwords are a problem on member and standalone servers, too, and the same solutions can be used to help solve the problem.
Service Management

Services are another area in which client computers can present security difficulties. Client computers come with several pre-enabled services, many of which can be disabled entirely. Even those services you choose to leave running, however, present security risks when configured to run as an over-privileged account or when configured to run as an account whose password is never changed. Of course, Windows doesn’t make it easy to keep service accounts properly configured, so you’ll need you use some creativity.

Unnecessary Services

Every service in Windows is necessary to someone—Microsoft didn’t include any services that do nothing all the time in every environment. By unnecessary services, I’m referring to services that provide features or capabilities that many environments don’t utilize. Why disable these services? History tells us that eventually a bug will be discovered in one of these services that will allow attackers to perform any number of heinous acts on the computer. By disabling services that you’re not utilizing, you’ll help prevent these services from becoming an attack vector in the future.

Disabling a service is easy. Simply right-click My Computer, select Manage, then open the Services node in the left-hand tree view. You can double-click any service to change its startup type to Disabled, and you’ll be able to stop the service if its running. Once set to Disabled, a service can’t be started unless its startup type is first changed to Automatic or Manual.

For even more security, uninstall the service if possible. For example, rather than just disabling Internet Information Services (IIS), uninstall it from the Add/Remove Windows Components utility in the Control Panel (accessed through Add/Remove Programs). Most built-in services can’t be removed in this fashion, but some can, and by removing the software you’ll eliminate the potential for someone to re-enable and start the service.

The following list of services—some of which are disabled by default—I recommend disabling (and, if possible, removing):

- Alerter—This service allows the computer to send and display certain types of alerts; primarily used with older software from the Windows NT days.
- Application Layer Gateway Service—This service is not required after Windows XP SP2 is installed.
- ClipBook—This service is an extension of the Windows Clipboard functionality and is disabled by default.
- Computer Browser—This service maintains a listing of network computers and resources; servers will typically provide this functionality, and clients shouldn’t typically run this service. If you have a good DNS infrastructure and your users aren’t accustomed to “browsing” the “network neighborhood,” disable this service on all machines.
- Error Reporting Service—This service provides a pop-up dialog box that offers to transmit errors and application crashes to Microsoft; it is unnecessary.
• FTP Publishing—This service is part of IIS. It is generally not appropriate for a client computer to be hosting an FTP site, so this service can be disabled and uninstalled.

• Human Interface Device Access—Usually disabled by default anyway, this service is necessary only for certain complex keyboards and other interface devices.

• IIS Admin—Part of IIS and rarely needed on client computers, this service can be disabled and uninstalled.

• Indexing Service—This service provides indexing of files on the local drive for faster searching; it is rarely used by most users and is therefore a good candidate for disabling.

• IPSec Services—This service is necessary only if you’re using IPSec or L2TP Virtual Private Networks (VPNs).

• Message Queuing—This service is necessary only for applications that utilize Microsoft Message Queue (MSMQ) services.

• Messenger—This service is not MSN Messenger or Windows Messenger; it is a separate service used with the NET SEND command and can almost always be disabled.

• MS Software Shadow Copy Provider—Microsoft Backup tries to use this service; the service is not usually necessary if you aren’t using Backup.

• Net Logon—This service is not usually required on a standalone system; it is required to log on to a domain controller.

• Network DDE—This service is not required by most systems.

• Network DDE DSDM—This service is not required by most systems.

• Network Location Awareness—This service is not required after Windows XP SP2 is installed.

• Network Provisioning Service—This service is used with domain controllers and XML configuration files; it is not required for standalone computers, but might be needed in a domain environment.

• Peer Name Resolution Protocol—This service is disabled (or removed) after Windows XP SP2 is installed; rarely needed and used primarily by IPv6.

• Peer Networking—This service is disabled (or removed) after Windows XP SP2 is installed; rarely needed and used primarily by IPv6.

• Peer Networking Group Authentication—This service is disabled (or removed) after Windows XP SP2 is installed; rarely needed and used primarily by IPv6.

• Peer Networking Identity Manager—This service is disabled (or removed) after Windows XP SP2 is installed; rarely needed and used primarily by IPv6.

• Performance Logs and Alerts—This service is rarely used on client computers and can be disabled; enable it if you specifically need to create performance logs and alerts.

• Portable Media Serial Number Service—This service is generally used only by Windows Media Player’s Digital Rights Management and can often be disabled with no ill effects.
• Remote Desktop Help Session Manager—If you don’t use Windows XP’s Remote Assistance feature, this service can be disabled.

• Remote Registry Service—This service provides remote access to the registry; if you don’t need that (keeping in mind that Windows Management Instrumentation—WMI—provides an alternative method for remotely accessing the registry), disable this service.

• Routing and Remote Access—This service is usually disabled by default because client computers don’t typically accept incoming connections.

• Secondary Logon—If you don’t utilize the “Run As” command to run applications under alternate credentials, disable this service.

• Security Center—This service monitors Automatic Updates, the Windows Firewall, and other features; disabling this service simply removes the ability for Windows to alert you when, say, your virus definitions are out of date (something your antivirus software will likely do for you on its own anyway).

• Server—This service is used for file and print sharing; if your client computers don’t share files and printers, disable this service. Doing so doesn’t stop users from connecting to shared files or printers on servers.

• Simple Mail Transport Protocol (SMTP)—This service is part of IIS and should usually be removed if you’re not using the machine as a mail server.

• Simple TCP/IP Services—This service is a rarely used minor TCP/IP service; it can usually be disabled.

• Smart Card—Not using smart cards? Disable this service.

• SNMP Service—If you’re not using SNMP, disable this service.

• SNMP Trap Service—Disable this service if you’re not using SNMP.

• SSDP Discovery Service—This service is used as part of Universal Plug-n-Play and detects and configures UPnP devices on a home network; it is rarely used in a corporate environment. MSN Messenger does rely on this service on certain types of networks to get outside the firewall.

• TCP/IP NetBIOS Helper Service—If you’re not using WINS, you can disable this service.

• TCP/IP Printer Server—This service provides TCP/IP-based print sharing and can usually be disabled on client computers.
• Telnet—This service is usually not appropriate for client computers and can be disabled.
• Uninterruptible Power Supply—It’s rare for a client computer to have a smart UPS—one that can shut down the computer if the UPS is on battery power and is running low; thus, this service can usually be disabled.
• Volume Shadow Copy—This service can generally be disabled on a client computer.
• WebClient—This service can be disabled and isn’t currently used by anything that I’m aware of on client computers.
• World Wide Web Publishing—Again, part of IIS, this service is not generally appropriate for a client.

So how do you go about enforcing your disabled service decisions across your enterprise? Group Policy is a start. As Figure 2.4 shows, you can use a GPO to enforce the startup type for any of the built-in services.

Figure 2.4: Disabling services through Group Policy.
Several services’ names changed in Windows XP SP2; be sure you’ve got the proper GPO templates on your domain controllers so that the list shown will reflect the version of Windows XP you’re using in your environment.

Service Logons and Passwords

Although Group Policy lets you decide which services will be allowed to run, it does nothing for helping you manage two important aspects of services:

- The account they will run under
- The password for that account

Many services, for example, are configured to run under the all-powerful Local System account; such is especially true on server computers on which additional services for SQL Server, Exchange Server, and other add-on applications are running. Even on client computers, however, you might want to alter the account that a service is using to reduce its permissions to a more reasonable level. More importantly, any service not running as Local System will be logging on using a password, and that password will need to be changed on a regular and fairly frequent basis, just like any user password.

If your company must remain compliant—for example, with the Sarbanes Oxley Act—and your company policy is to change user passwords every 45 days, you must include the often-neglected service accounts if you are to maintain regulatory compliance.

Changing a service’s password involves two steps: Changing the password of the user account (which, if it’s a local account, can be a time-consuming task without some kind of tool to help out), then telling the service itself to use the new password. That latter step can be exceedingly painful, especially if the service is installed on many computers.

Obviously, this area is where many administrators will write (or download) a script of some kind to do the job. Although this solution is okay, it typically assumes that you know which computers are running the service in question. To be on the safe side, you really need a tool that can first find all computers running the service, then reconfigure the service’s password. ScriptLogic Service Explorer, which Figure 2.5 shows, has a search function that will search entire domains or workgroups for specified services, then allow you to configure those services en masse.
Figure 2.5: Searching for the Application Layer Gateway Service.

Service Explorer has several helpful built-in searches, as well, such as one that looks for non-Microsoft services and another that displays all services that use a particular user account to log on. This type of search is useful when you’re changing a password: Find every service actually using the account in question!

A similar tool, Lieberman Software’s Service Account Manager, works similarly. As Figure 2.6 shows, Service Account Manager provides a single view of all services on a given machine. It can also locate machines running a particular service, and when updating a service’s logon password, it can update the locally cached credentials for the service, allowing it to log on and continue running even if the computer temporarily loses connectivity with a domain controller (for services logging in under a domain account).
The bottom line is that managing services is perhaps one of the most-overlooked client security problems, and there are tools that can help you solve the problem very, very easily. Getting your services locked down and your service logon passwords under control is a great step toward a more secure Windows enterprise.
Local Firewalls

I’ll risk starting a firestorm of debate with this statement: Every Windows client computer should have a local firewall. Now that I’ve said it, let me defend myself, because I know the topic of local firewalls is one that creates a lot of tension in the Windows administrative community.

Some administrators hate local firewalls, and for good reason. They definitely increase the administrative burden client computers represent. You’ll need to be more concerned with what client computers are doing so that you can configure the firewall appropriately. I don’t think that “additional administrative burden,” however, is a good excuse for lax security. The fact is that most attacks target client computer vulnerabilities; because you can never tell what vulnerabilities might be lurking in Windows or your other corporate software, a firewall provides a good, solid line of defense. Keep in mind that most attacks come from within your network, so don’t think that the corporate firewall is a perfect defense that obviates the need for a per-client defensive mechanism.

Microsoft’s Windows Firewall, installed in Windows XP SP2, is a decent client-side firewall; other client-side firewalls are available from several companies. Windows Firewall has the benefit of being centrally configured through Group Policy: You can turn it on and off, configure port exceptions to allow incoming traffic, and so forth. Because most client computers don’t need to accept incoming connections (excepting, of course, replies to network traffic that originated on the client; replies are allowed by default), you can often just configure the firewall to be on and leave it at that.

If your domain controllers aren’t showing the Windows Firewall Group Policy settings, you can add them by downloading the appropriate SP2 ADM files from http://www.microsoft.com/downloads/details.aspx?FamilyID=92759d4b-7112-4b6c-ad4a-bbf3802a5c9b&displaylang=en#filelist.

My complaint about Group Policy is that it is not quite granular enough in its application. GPOs can be linked to OUs, domains, or sites; the application of a particular GPO can be blocked at any of those levels, as well. With Windows XP and Windows Server 2003 (WS2K3) systems, application can be made a bit more granular through the use of WMI filters. However, you can’t easily, for example, apply a GPO only to members of a certain group who have a particular software application running on their computer. ScriptLogic Desktop Authority, however, can apply Windows Firewall settings at that kind of granularity. For example, Figure 2.7 shows, I’ve created a Desktop Authority setting that enables the Windows Firewall and creates a port exception allowing incoming traffic on TCP port 80 (strictly as a demonstration; few client computers would actually need such an exception).
Figure 2.7: Configuring a firewall setting in Desktop Authority.

I can restrict application of this setting to only those computers that are not members of the Administrator PCs group. In other words, administrators’ PCs won’t have this policy applied; a perfectly reasonable requirement in many environments in which administrators run software that ordinary users never will. A great many rules can be applied to the setting, such as the type of machine (desktop, laptop, tablet PC, and so forth), the OS, the type of connection, and so forth.

Although Windows Firewall was first introduced in Windows XP SP2 and you will most commonly use it on client computers, WS2K3 SP1 also contains the Windows Firewall and makes it available on server computers.
NTFS Permissions

Consistent file permissions are crucial to enterprise security. As I’ve already mentioned, client computers often contain many confidential files, but client computers are often perceived as being less critical from a file permissions viewpoint. This position is dangerous: Imagine the damage that could be caused if someone swiped a company laptop in an airport waiting lounge, for example.

Companies dealing with regulatory compliance issues—such as the Health Insurance Portability and Accountability Act (HIPAA), for example—should be very concerned about the security of files on client computers. Some organizations might want to keep files from being stored on client computers—a topic I’ll address in the next few sections. However, sometimes allowing users to keep local copies of files—especially on laptops—is unavoidable. In those situations, having the correct file permissions in place is critical to maintaining your compliance.

Windows security templates can be used to create a consistent NTFS permissions structure. For example, the Setup Security.inf security template—shown opened in the Security Templates console in Figure 2.8—applies the starting security permissions for the entire OS.

Figure 2.8: Using security templates to manage NTFS permissions.
You can create your own security templates, import them into a GPO, and apply them to a domain, site, or OU. Templates are a decent way to configure consistent NTFS permissions for a particular folder structure across a large number of computers. However, templates are far from a perfect solution. For example, they require all targeted computers to have an identical folder structure (at least within the folder structure you’re defining in the template), which isn’t always the case. Further, templates provide no reporting capability, which would allow you to easily verify the NTFS permissions applied to a given file or folder.

Third-party tools can, however, provide a robust level of reporting and help manage security more easily. BindView Corporation makes a suite of products designed to help organizations better meet regulatory and industry standards, including security permissions and auditing settings on files.

ScriptLogic Enterprise Security Reporter helps you effectively manage security and can also provide robust reports for client-level security. Although Enterprise Security Reporter is intended primarily for reporting on server-based security, many of its functions can be useful for client-based security as well. In a compliance environment, you might even be required to provide these types of reports for your client computers. The tool starts by loading security information from targeted computers into a SQL Server database, which allows you to then instantly obtain security reports, such as reports on which users have specific permissions under a given folder hierarchy. Figure 2.9 shows a sample report, listing the users that have permissions under a specified folder. This type of report is excellent for compliance management, because it lets you quickly verify that only the proper users have permissions on folders known to contain confidential information.

![Figure 2.9: Viewing permissions assigned to a specified folder.](image-url)
For a more interactive security tool, ScriptLogic Security Explorer allows you to create **scopes**, which are collections of targeted security elements—including, for example, folders. Figure 2.10 shows Security Explorer examining the permissions on a folder that has been added to a scope; this folder might be an application data folder, for example.

![Security Explorer](image)

*Figure 2.10: Viewing security permissions through Security Explorer.*

Once you have a scope established, you can conduct searches on it, modify its permissions, and so forth. For example, in Figure 2.11, I’m conducting a search on a scope named Clients. I might look for anything within the scope that assigns permissions to the Everyone group, or for permissions assigned to a particular user or group. I can search permissions on files, folders, and subfolders within the scope, and I can restrict the search to a particular set of permissions. By using this powerful search mechanism, I can quickly locate undesirable permissions, then use Security Explorer to remove or modify them.
Figure 2.11: Searching a scope in Security Explorer.

These tools can all help you maintain more consistent permissions. Of course, don’t forget about freely available tools for managing permissions, such as Windows’ built-in Cacls.exe command-line tool and the more flexible Windows resource kit tool, XCacls.exe. Although less suitable for administration of multiple computers, these tools can allow you to quickly reconfigure security permissions in a folder hierarchy on a single machine, and they can be used in a batch file to make it easier to make changes across multiple machines at once.

However, bear in mind that maintaining consistent permissions across multiple client computers will always be difficult. A better idea, if possible, is to simply get the files off of the client computers entirely.
Folder Redirection

The idea behind folder redirection is simple, and is illustrated in Figure 2.12. Users access what appears to be a local folder, but that access is redirected to a server-based folder. Users typically never realize that the files are located on a server rather than their local computers. The benefit is that files can be centrally secured and audited on the server, eliminating the need to worry about consistent security on the client. The client doesn’t actually contain the files, and therefore doesn’t need special security considerations. Files can also be more easily backed up and restored on the server than on a client.

Figure 2.12: Folder redirection keeps folders and files centrally located.

There is a downside to folder redirection for portable computers because users think they have all the files locally and don’t realize folder redirection is in effect. When they disconnect from the network, their files “vanish.” Using Windows’ Offline Files feature can help mitigate this problem.

Folder redirection for certain special folders—most commonly, users’ profile folder, which contains the My Documents folder—can be accomplished through Group Policy. Larger organizations often need to redirect folders based on the user so that different users’ redirected folders can be housed on various file servers, providing sufficient storage for everyone. Group Policy accomplishes this task most easily if you can have file servers that correspond with AD OUs. In this case, you create a unique GPO for each OU and redirect that OU’s users (or rather, those users’ folders) to a particular server. Otherwise, you might need to use a technology such as Windows Distributed File System (DFS), which can provide a non-server-centric view of the network, allowing users’ folders to be redirected to an arbitrary server, as Figure 2.13 illustrates.
Another technique is to use a desktop configuration tool such as ScriptLogic Desktop Authority. As Figure 2.14 shows, this tool can redirect many different shell folders to an arbitrary location. It can also copy any files that already exist locally in the to-be-redirected-folder to the new location, ensuring a transparent cutover to the redirection scheme. Desktop Authority 6.05 can redirect the following folders:

- Start menu folder
- Programs folder
- Startup folder
- Desktop folder
- Favorites (Internet Explorer bookmarks) folder
- Personal (My Documents) folder
- My Pictures folder
- Cookies folder
- History folder
- Recent folder
- Send To folder
- Temporary Internet Files folder
Desktop Authority also provides for more granular application of settings than Group Policy offers. For example, this setting might be applied to all desktop computers, but not to portable computers (for whom redirection can be problematic because the computer isn’t always on the network).

![Figure 2.14: Redirecting the My Documents folder.](image)

Folder redirection is a crucial security tactic, allowing you to maintain ease-of-use for your users while consolidating files onto more easily secured and easily audited file servers. Folder redirection can also help enforce system configurations. For example, if all users’ desktop folders are redirected to a single shared location, and users are not given write permissions to that location, then all users will have a consistent, locked desktop configuration. Such a configuration can make it more difficult for users to introduce external software—such as viruses—by locking down some portion of the file system where Web browsers and other applications try to save files.
Removable Storage

Organizations have long sought to lock down removable media, a key means of introducing unwanted software into the environment and for removing confidential information from the environment. In the past, organizations might order computers without floppy drives or might restrict the use of optical media burners. However, removable media today is ubiquitous, with FireWire/IEEE1394 and USB devices making it easier for users to take data in and out of the environment without notice. Third-party tools currently provide the only reliable means of locking down these external, removable storage options.

Why bother locking down USB flash drives, for example? Because most removable media support only the FAT, FAT32, or CDFS file systems, none of which support security permissions. Thus, removable media not only represents an opportunity to introduce unwanted software and to remove confidential data but also ensures that any data removed from the environment will be completely unsecured. Although some removable media offers security options such as encryption, there is no centralized means of enforcing the use of such features, making it less likely that users will do so.

SecureWave Sanctuary is designed for device access control. Devices are categorized—digital cameras (which have onboard storage), optical burners, smart card readers, flash drives, and so forth—and, by default, disabled. You can “white list” allowed devices, such as scanners or modems, and leave all other devices disabled. Users are unable to install the devices under Windows, meaning they are unable to use disallowed devices to bring data in or out of the environment. Device access can be granted on a temporary, per-user basis if necessary. You can even allow optical drives to function, but provide a list of allowable media, ensuring that users can run authorized software but not introduce new software into the environment.

Another similar package is GFiLANguard Portable Storage Control (PSC), which focuses exclusively on portable storage such as USB flash drives. It addresses almost all forms of portable storage, including flash drives, MP3 players and smartphones, digital cameras, CDs, floppies, and so forth. As Figure 2.15 shows, device permissions can be mapped to AD groups, helping to minimize security management overhead. For example, you might create groups that represent allowed devices, then simply add users to the groups on an as-needed basis.
Don’t forget all the ways in which data can leave a computer or enter it. Your network is one obvious way, but that’s something you can control. Any portable device with memory—such as a digital camera—is a possibility. Also keep in mind Bluetooth- and infrared-accessible devices, such as PDAs and smart phones, and be sure to control them appropriately.

Controlling access to removable storage will help make your environment more secure by reducing the ways in which information can leave your network and reducing the ways in which unwanted software can enter your network.
Local System Permissions

Local system permissions are the final area covered in this chapter. Consider Cmd.exe, a file that is usually located in C:\Windows\System32. Figure 2.16 shows the file permissions on Cmd.exe on a Windows XP Professional computer that has been upgraded to SP2.

Notice that the SYSTEM account has Full Control. Why would the system itself need to open a new command-line window? I typically remove the SYSTEM account from this and many other files in the file system.
Also notice that the Internet Guest Account has permissions to run Cmd.exe. Anonymous users have the ability to open a command-line window and execute commands. Spend some time investigating the default permissions on the many files and folders lurking around in Windows and to apply more sensible defaults. Some other files you might want to investigate include:

- Command.com
- Ftp.exe
- Tftp.exe
- Telnet.exe
- WScript.exe
- CScript.exe
- Net.exe

Don’t try to delete these files; most are under Windows File Protection and will be replaced eventually (by a service pack, if nothing else). Instead, modify the permissions on these files so that only appropriate users—real users, not SYSTEM—can execute them.

The idea is that these types of all-powerful utilities can create significant havoc if an attacker gains access to them. Reduce the likelihood of that happening by removing access from any account that doesn’t absolutely need to have it.

**Summary**

Client computers represent a significant security risk in many organizations simply because they’re rarely as controlled or as well-configured as servers. This chapter has introduced you to some of the major client vulnerabilities and given you some tips on how to lock them down appropriately. One way to get a better handle on client security is to think about the entire life cycle data takes in your organization—from the server, across the network, to the client, to portable devices, and so forth. Thinking about that life cycle will help you better implement appropriate levels of security at each point in the cycle.

The next chapter will focus on a topic that affects both clients and servers—the software built-in to Windows that presents major vulnerabilities. Often called “middleware,” applications such as Internet Explorer (IE), Windows Media Player, and other applications have a reputation for security problems. I’ll show you some ways in which those problems can be addressed and mitigated.
Chapter 3: Using Alternative Software to Reduce Your Attack Surface

*Attack surface* is an IT security term that refers to the number of ways in which a computer can be attacked. For example, a computer running an older operating system (OS) such as MS-DOS has a relatively low attack surface, because the code base for MS-DOS is so much smaller than that of current OSs and few of today’s attack techniques would work against it. A computer running Windows XP Professional, in contrast, has a relatively large attack surface simply because it is such a large, complex OS.

As the previous chapter explored, one way to reduce attack surface is to uninstall software that isn’t being used, such as unnecessary services. Local firewalls, such as the Windows Firewall, can also reduce your attack surface by blocking attacks’ access to specific applications (such as a local Web server). However, it’s still possible in many cases for attacks to get through. Suppose that a Windows XP Professional user has the Windows Firewall turned on full-force, and that user attempts to visit a Web site. The firewall will allow the user to visit the site because the traffic is originating locally and that type of traffic isn’t blocked by the firewall. The Web site’s content, which comes back as a reply to the locally generated traffic, is also allowed. If that content contains an attack—say a virus or other malware—the attack is allowed into the computer where it can do its damage.

Unfortunately, much of Windows XP’s bundled software is rife with security vulnerabilities, allowing numerous types of attacks to be effective (this fact applies to earlier versions of Windows, as well). Thus, a further technique of reducing your attack surface—beyond removing unused software and using a local firewall—is to replace this bundled software with less-vulnerable alternatives.

The Bundled Software

The bundled software that this chapter explores is commonly referred to by Microsoft as *middleware*—those applications that Microsoft has controversially bundled into the OS rather than offering them as standalone packages. These applications include:

- Internet Explorer (IE)
- Windows Messenger (this reference doesn’t include the similar MSN Messenger, which isn’t bundled with Windows and can be easily uninstalled)
- Windows Media Player
- Outlook Express

There is some discrepancy regarding the definition of *middleware*. In addition to the version used by Microsoft, which is the definition used for this guide, the term middleware is also used to refer to separate products that serve as the glue between two applications.
These applications are worthy of focus because they are installed by default, are somewhat difficult to actually remove if you’re not using them, and provide functionality that most users require. For example, most users need a Web browser and an email client; because IE and Outlook Express are bundled, users might be using those applications to provide the functionality they need. Even if these applications are not being used, however, they present a potential security risk.

Dealing with the issue of bundled software is complex, most particularly in the case of IE. To tackle this topic, let’s first explore the specific functionality that you will be missing if you eliminate these bundled applications. Second, we’ll discover how to actually remove or disable them to the greatest degree possible. Finally, let’s look at alternative applications that provide most, if not all, of the same functionality, and any weaknesses (especially in terms of enterprise management) that the alternatives present.

<table>
<thead>
<tr>
<th>What About Alternatives to Other Applications?</th>
</tr>
</thead>
<tbody>
<tr>
<td>This chapter isn’t intended to be Microsoft-centric; it simply focuses on those applications that are preinstalled by Microsoft, less than straightforward to remove, and offer must-have functionality by virtually all modern users. Certainly, there are alternatives to nearly every application that Microsoft offers, and the manufacturers of those alternatives will be more than happy to tell you how their products work, if you’re interested. We’ll focus on the Microsoft bundled software because its combination of preinstallation, lack of straightforward removal options, and mission-critical functionality places this software into a special category all its own.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>What Will You Miss?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft’s bundled applications do, in most cases, provide unique functionality, both in terms of end-user functionality and enterprise manageability. Doing away with the bundled applications will, in most cases, remove that unique functionality. In some cases, this removal might not affect users—Windows Media Player’s unique ability to play Windows Media Video (WMV) files, for example, might not matter to your organization; thus, losing that capability might not be a “loss” to you at all. It is important to simply know what functionality is going to go missing when you eliminate these bundled applications so that you can make an informed decision about whether losing that functionality is acceptable.</td>
</tr>
</tbody>
</table>

There is also a middle ground where you don’t eliminate the bundled applications but simply stop using them as the default. For example, you might choose to use an alternative Web browser as your users’ default Web browser. Doing so would help eliminate many of the vulnerabilities associated with IE because those vulnerabilities only work if IE is being used to access Web pages containing an exploit. This method doesn’t require you to remove IE; you can still leave it available to your users for tasks that require IE’s unique functionality.

Later in the chapter, I’ll explore this middle ground scenario.
IE has been host to so many security vulnerabilities over the years that it is one of the first bundled applications that enterprises look at for potential removal. Doing so can be difficult. Although few Web sites used by corporate users require IE’s unique features, many intranet applications were built around those features.

In terms of user functionality, IE doesn’t offer much that can’t be replaced by an alternative browser. In fact, most alternatives offer superior user interface (UI) features—such as tabbed browsing—that IE hasn’t offered.

Microsoft recently announced IE 7.0, which will ship as an update to Windows XP prior to the release of Windows Longhorn; version 7.0 might offer tabbed browsing and other UI enhancements.

Most users are receptive to alternative browsers and do well with them; the alternatives work similarly enough to the familiar IE that there is practically no learning curve. There is a small amount of functionality that isn’t exactly unique to IE but that IE handles slightly differently than the alternatives. For example, in Figure 3.1 you see a portion of a Web page that is using a Dynamic HTML (DHTML)-generated drop-down menu. Look carefully, and you’ll see that the right side of the menu’s black background fades into transparency, allowing the underlying page to be glimpsed. This visual effect looks nice, and is essentially unique to IE (at least for now) because it is using Microsoft-designed extensions to Cascading Style Sheets. Although an alternative browser can duplicate the operation of the drop-down menu, it might not duplicate this transparency effect. However, this shortcoming is something few users will notice or mind in most cases.

In terms of developer functionality—that is, functionality that developers of Web applications rely on—IE is rich with unique features. Although the most recent alternative browsers do a good job of replacing most IE-specific functionality, there are three major pieces of functionality that are still unique to IE: VBScript, ActiveX, and XML.
Many Web applications rely on client-side scripts to create robust functionality in their Web applications. For example, the drop-down menu in Figure 3.1 was created by using client-side scripting, a technique referred to as DHTML. Client-side scripting requires a scripting language; while most developers will elect to use a language such as JavaScript—which is compatible with most browsers—some developers will use VBScript, a Microsoft-proprietary language that pretty much only works with IE.

### JavaScript vs. JScript vs. ECMAScript

For accuracy’s sake, I want to point out that JavaScript is a language originally developed by Netscape. It does bear a passing resemblance to Sun Microsystems’ Java language, but it’s a language in its own right. Ecma International, an industry association focused on IT and communications standardization, adopted JavaScript as a standard language. The specific implementation of JavaScript that was adopted by Ecma is known as ECMAScript (you’ll find the specification at [http://www.ecma-international.org/publications/standards/Ecma-262.htm](http://www.ecma-international.org/publications/standards/Ecma-262.htm)).

JScript is the language that is built-in to IE; this Microsoft-developed scripting language is compliant with the ECMAScript standard. Although IE recognizes the use of JavaScript as a client-side language, IE actually executes JavaScript by using the JScript language engine. The practical upshot of all this is that there are three different names for what amounts to the same thing, once you discard the legal and trademark issues involved.

Ditching IE means losing VBScript capabilities. This is where the middle ground scenario, which I’ll discuss in more detail later in this chapter, comes into play: Most instances where you will need VBScript relate to intranet applications, and presumably (or at least hopefully) you can trust your own intranet applications not to launch attacks against your client computers. That makes IE—and its support for VBScript—safe for internal use where you need it, and still makes it feasible to use an alternative browser for Internet work, where VBScript is less likely to be encountered.

---

You still might run into Internet situations in which VBScript—and therefore, IE—is needed. Extranets are a good example. Because extranets are often developed as in-house applications, VBScript is more likely to be in use. For example, although much of Microsoft’s public Microsoft.com Web site is browseable by non-IE browsers, many of their extranet applications aren’t. Microsoft’s invoicing Web site, used by Microsoft’s vendors to submit invoices to the company, requires a browser that supports IE because it uses VBScript.

By using VBScript in an extranet application, you are essentially forcing your partner companies to use a particular piece of software—in this case, IE—to access your extranet, which is a less-than-ideal business practice. Extranets should be the first applications targeted for revision to remove IE-specific features so that your business partners can make their own decisions about what software to use.

ActiveX is another technology supported exclusively by IE. ActiveX allows small, self-contained applications—ActiveX controls—to be downloaded into IE and executed within the IE window. The industry alternative to ActiveX is Java, which provides similar functionality albeit through a radically different technological approach. In other words, while rewriting a Web application to use ECMAScript instead of VBScript is pretty straightforward, rewriting an ActiveX control as a Java applet is anything but straightforward.
Very few legitimate Internet sites rely entirely on ActiveX controls, primarily because ActiveX controls are the source of many of IE’s vulnerabilities (there are some notable exceptions, but many sites simply fear using ActiveX controls because of their poor reputation). Losing the ability to use ActiveX controls is, in fact, seen as a benefit to many organizations (to organizations that don’t have any end-user needs that would require ActiveX, but making this determination in large organizations can be a difficult task).

However, there are drawbacks to losing ActiveX, primarily in relation to Microsoft-based Web sites and functionality. Perhaps the most obvious example is the Windows Update Web site, used to check Windows computers for the latest patches and software updates. The software that performs those checks and downloads and installs new software is an ActiveX control. Try to access Windows Update without an ActiveX-capable browser (in other words, without IE), and you will receive a message like the one that Figure 3.2 shows.

![Figure 3.2: You need IE to access Windows Update.](image)

Workarounds are available and numerous. For example, you could configure your Windows 2000 (Win2K) and Windows XP computers to use Automatic Updates, which will download critical updates without using IE. You could also deploy Microsoft Software Update Services (SUS 1.0, or its successor, which is called Windows Update Services—WUS), which works with Automatic Updates to deploy patches and other updates without the use of IE. However, this workaround applies only to the most obvious example of ActiveX usage; if you have business applications that rely on IE’s support for ActiveX, you might need to consider a middle ground solution in which you retain IE for at least some purposes.
A final unique feature in IE is its XML support, which allows the browser to perform client-side manipulation of data. Perhaps the best use of this technology is seen in Exchange Server 2003’s Outlook Web Access (OWA), where the Web interface is able to closely resemble the Outlook application. Users can sort their email, for example, by clicking a column, and the browser doesn’t need to reload the page; it simply uses IE’s XML support to redisplay the page client-side for a faster user experience. Although OWA provides compatibility for other browsers, the user experience is less true to the full Outlook application and feels more like a Web site than an application. In some cases (such as with OWA), giving up IE means giving up a rich user experience; in other instances (such as intranet applications designed for IE), the lack of this robust XML support in an alternative browser might make the alternative browser unfeasible or unusable.

What About IE 7.0?

In the first quarter of 2005, Microsoft announced IE 7.0, a new version of IE that would ship separately from Windows Longhorn (the next version of Windows itself), despite earlier announcements that IE would not ship a new version until the one bundled in Longhorn. Microsoft is touting IE 7.0 as a more secure Web browser; certainly, it has a host of new security features that build upon those added in Windows XP Service Pack 2 (SP2). However, for some organizations, Microsoft’s definition of “more secure” might differ from their own.

Microsoft’s approach to securing IE—an approach seen in Windows XP SP2 and used in IE 7.0—has primarily been to partially or conditionally block access to potentially vulnerable features or content. Many organizations find this approach to be insufficient and feel that Microsoft should remove the potentially vulnerable features. For example, allowing the user to block access to ActiveX controls isn’t as secure as removing support for ActiveX controls completely (thus ensuring that ActiveX controls can’t be “unblocked”). That backward step in functionality would create significant compatibility issues, however, which is why it is an approach that Microsoft is understandably hesitant to take.

Another problem is that these security updates to IE—in terms of service packs and new versions—are available only to Windows XP, which is not universally deployed. Users of Win2K or Windows 98, for example, must still deal with the less-secure IE they have always had.

Ultimately, Microsoft’s updates to IE do (and will continue to) make it less susceptible to attack, while preserving the unique functionality that IE offers. However, alternative browsers continue to offer a more secure alternative simply by not having the functionality that is so often exploited. Additionally, because alternative browsers are less tightly integrated with Windows itself, any vulnerabilities in those browsers will likely be a problem only for the browser—unlike IE, through which a browser-based vulnerability can open into the OS as a whole.

Outlook Express

Outlook Express is a Post Office Protocol 3 (POP3) and Internet Messaging Application Protocol (IMAP) client—in short, an email application. It also offers newsreader functionality for participating in Internet (USENET) newsgroups.

The market is flooded with alternatives to Outlook Express, ranging from freely available clients such as certain editions of Eudora to more robust applications. Certainly, Outlook Express does not offer the robust functionality unique to the full Outlook application (which is a part of the Microsoft Office suite). The one piece of nearly unique functionality Outlook Express does offer is security-related—digital signatures and message encryption. Although Outlook Express isn’t the only mail client to support digital certificates (the enabling technology behind signatures and encryption), it is one of the few Windows-compatible clients to do so.
Perhaps more important for this discussion, however, is the fact that very few organizations rely on Outlook Express. Although alternatives certainly exist, most organizations already have an alternative—such as Outlook, Lotus Notes, or some other messaging client—in place; they simply need to remove Outlook Express to reduce their attack surface.

### If It Ain’t Broke, Why Fix It?

The following scenario provides an example of why you might want to remove Outlook Express even if you are not using it—suppose you use Lotus Notes and one of your users receives a message with a file attachment. That file attachment has an .eml extension, which by default will open in Outlook Express. If Outlook Express is installed and available, double-clicking the attachment in Notes might open Outlook Express! Thus, Outlook Express could become “in use” even if you’re supposedly not using it.

Why would this be a problem? Outlook Express has been subject to several attacks and exploits. For example, the aforementioned .eml attachment might contain a JPEG graphic that exploits a vulnerability in Microsoft’s Graphic Device Interface (GDI+) software—a core part of Windows—to crash the computer or execute code. Because Outlook Express is integrated with Windows, Outlook Express uses Windows’ GDI+ to display the JPEG graphic, thus making the exploit work. (The GDI+ vulnerability has been patched, but organizations have been notoriously inefficient in ensuring that every copy of GDI+ on their computers receives the patch).

### Windows Media Player

There is certainly no shortage of media players on the market, but while all of them provide support for industry-standard file formats—such as MP3 audio or MPEG video—they also come with their own proprietary formats. In the case of Windows Media Player, the supported Windows Media formats include WMA, WMV, ASF, and so forth. Bottom line: If you need to access these media formats, you need Windows Media Player.

You might seek to eliminate Windows Media Player by converting any corporate content into other formats, such as Real Media or Apple QuickTime. Certainly many public Web sites offer media in one of these formats along with Windows Media Player, so converting your corporate content might cover the majority of circumstances in which Windows Media Player might need to be used—except, however, when it comes to protected content.

Windows Media has established a significant market lead in Digital Rights Management (DRM). Apple QuickTime doesn’t currently offer DRM for video content and doesn’t provide third parties with a way to protect audio content. Real Networks offers DRM for Real Media but requires you to purchase a licensing server in order to use the DRM features. Windows Media’s DRM can be deployed essentially free of charge because the DRM software development kit is freely available (although you do have to sign a license with Microsoft to obtain it). If your company is dealing with DRM-protected media content, the odds are that you will need to continue using Windows Media Player.
**Windows Messenger**

There are two Microsoft instant messaging clients: Windows Messenger, which comes bundled with Windows, and MSN Messenger, which is downloadable from Microsoft’s MSN Web site. Both are similar in look and operation and provide access to Microsoft’s instant messaging network. Windows Messenger can also connect to other instant messaging networks, such as those created by Microsoft Live Communications Server or Exchange 2000 Server. MSN Messenger can be easily uninstalled if you don’t want to use it; Windows Messenger is somewhat more difficult.

When I speak of alternatives to Windows Messenger, I’m referring to instant messaging clients that can connect to Microsoft’s public instant messaging network, replacing Windows Messenger’s functionality. I am not referring to alternative instant messaging networks such as America Online or Yahoo. Although these provide useful functionality and are indeed alternative networks, they do not replace the functionality of Windows Messenger by allowing you to connect to Microsoft’s instant messaging network.

If your users only use public instant messaging networks, they don’t lose much by giving up Windows Messenger and using an alternative. If your users rely on an internal instant messaging network (increasingly common in corporate environments), an alternative client might not be able to connect. Instead, you might need to consider an alternative internal instant messaging product, which will come complete with its own server component as well as a new client. Moving to an alternative instant messaging network may or may not change the feature set to which your users have access—some products offer more functionality than Windows Messenger; others offer less.

**Removing the Bundled Software**

Since the case with the United States Department of Justice in 2003, Microsoft has offered a Program Access and Defaults application, which allows you to set the application used for certain middleware functions. As Figure 3.3 shows, this application also lets you control access and defaults for Web browsing, email, media playing, instant messaging, and your Java virtual machine (JVM).

This chapter doesn’t discuss Microsoft’s JVM because Microsoft is no longer developing the JVM. As a practical matter, you should install Sun Microsystems’ JVM and use it to achieve the most recent functionality.
This application allows you to determine two things:

- Which application will be used by default for these middleware applications
- Which applications will be accessible by users

For example, you can see in the figure that I’ve configured Mozilla Firefox to be accessible and disabled access to IE. Firefox is configured as my default browser (which you don’t see because this application defaults to “Use my current Web browser,” meaning it won’t change whatever the current setting is but also won’t show you what that current setting is).
This application is, however, misleading. Reading it, you would expect that IE would be turned off, shut down, and possibly even uninstalled. Nothing could be further from the truth: Other applications can launch IE, and, indeed, I can do so manually simply by opening Run from the Start menu, typing

    iexplore

and clicking OK. The “enable access” check box, when cleared, simply removes the IE icon from the Start menu—hardly a security precaution. The application does allude to this “enable” functionality in its description at the top of the window, but doesn’t make it clear that “other locations” still allows the application to be executed. In addition, disabling access to an application only removes the default icons that Windows creates for them; you can still easily create your own icons even if access to the application is “disabled.” From a security perspective, this application is fairly useless. So how do you go about removing bundled applications if you don’t want them?

**IE**

IE is perhaps the toughest bundled application to remove—assuming for the moment that you do want to remove it and not go with a middle ground solution, which I’ll discuss later in this chapter.

It is important to voice some cautions and caveats here. Understand that IE is composed of several different bits, some of which are absolutely critical to the operation of many applications. For example, the Microsoft HTML Rendering Engine (MSHTML) is the bit of IE that turns HTML code into a visible Web page; removing MSHTML will break plenty of applications. Iexplore.exe is actually just a wrapper around these several components; removing it doesn’t actually remove most of the software that contains IE’s vulnerabilities.

Bottom line: Removing any portion of IE might break some applications. Removing anything less than all of IE might leave some vulnerabilities in place (if the components you leave in place contain vulnerabilities, of course). So you should carefully test any IE removal with all of your company’s applications to determine if anything will be affected.

IE is heavily integrated with Windows and isn’t something you can safely just delete on your own. You need an application that understands how IE is built and can safely extract it. One such tool is IERadicator ([http://www.litepc.com/ieradicator.html](http://www.litepc.com/ieradicator.html)), which can de-register and remove most of the IE components, leaving the ones most commonly used by other applications. The application works with all versions of Windows earlier than Win2K SP2; for Win2K SP2 and later, including Windows XP, use the same company’s XPlite and 2000lite software ([http://www.litepc.com/xplite.html](http://www.litepc.com/xplite.html)).

I need to stress again, however, the importance of testing this process on a representative computer to make sure that none of your applications are adversely affected.

Don’t think for a second that going into the Control Panel’s Add or Remove Programs utility, selecting Add/Remove Windows Components, and clearing the IE check box will remove anything. As you can see in Figure 3.4, I’ve “removed” IE, yet you can still see it up and running.
Figure 3.4: Removing IE only removed its icon, not the application.

I’m not going to try and tell you that obtaining a tool such as IERadicator or XPlite is the only way to remove IE; obviously, anything an application can do for you, you can do yourself. But IE’s roots go deep, and the detailed, step-by-step actions that would be required to remove any part of it with any kind of safety isn’t something I’d trust myself to do manually.

Be aware that installing a service pack or even certain updates from Microsoft might reinstall portions of IE; test these updates to check their effect and determine whether you will need to subsequently re-un-install IE after applying the update.

Not many companies are able to remove IE without missing some of its functionality. Most companies continue to need it for one reason or another. In many cases, intranet applications requiring ActiveX controls of VBScript were the reason; OWA is another commonly cited reason for keeping IE (OWA in IE really is amazing). That doesn’t mean you can’t reduce your attack surface by using an alternative browser, though; later in this chapter, I’ll discuss coexistence scenarios in which you use IE only where doing so is advantageous, and use an alternative elsewhere.
**Outlook Express**

Like IE, Outlook Express shows up in the Add/Remove Windows Components dialog box, so you would think that clearing the check box and clicking OK would remove it. Not so: Look at Figure 3.5 in which the check box has been cleared, yet Outlook Express is clearly alive and well.

![Windows Components Wizard](image)

**Figure 3.5: “Uninstalling” Outlook Express only removes its icons from the Start menu.**

Truly uninstalling Outlook Express requires more effort, and you will need to be a local administrator in order to pull it off. You will be renaming several files and folders. What you rename them doesn’t matter, so long as you rename them. Start with:

- C:\Program Files\Common Files\Microsoft Shares\Stationery
- C:\Documents and Settings\username\Application Data\Identities or C:\Documents and Settings\username\Local Settings\Application Data\Identities
- C:\Documents and Settings\username\Application Data\Microsoft\Address Book or C:\Documents and Settings\username\Local Settings\Application Data\Address Book
You will need to delete several registry keys, as well.

The usual warnings about registry editing apply here: If you mess up the registry, you may mess up your computer beyond repair.

- HKEY_LOCAL_MACHINE\Software\Microsoft\Outlook Express
- HKEY_LOCAL_MACHINE\Software\Microsoft\WAB
- HKEY_CURRENT_USER\Identities
- HKEY_CURRENT_USER\Software\Microsoft\Outlook Express
- HKEY_CURRENT_USER\Software\Microsoft\WAB
- HKEY_LOCAL_MACHINE \Software\Microsoft\Active Setup\Installed Components\{44BBA840-CC51-11CF-AAFA-00AA00B6015C}
- HKEY_LOCAL_MACHINE \Software\Microsoft\Active Setup\Installed Components\{7790769C-0471-11D2-AF11-00C04FA35D02}

Now you have a bunch of files to rename—I recommend leaving the filenames and adding “.old” to the end (for example, Inetcomm.dll becomes Inetcomm.dll.old) so that you can easily put the files back if desired. You might want to use Windows Explorer’s Search function to find these files. Be aware that most of them will also exist in C:\Windows\System32\Dllcache; be sure to rename both copies to have the same new name:

- Inetcomm.dll
- Msoeacct.dll
- Msot2.dll
- Msom.dll
- Msoes.dll
- Msinn.exe
- Oeimport.dll
- Oemiglib.dll
- Oemig50.exe
- Setup50.exe
• Wab.exe
• Wabfind.dll
• Wabimp.dll
• Wabmig.exe
• Csapi3t1.dll
• Directdb.dll
• Wab32.dll
• Wab32res.dll

You might receive a prompt about Windows File Protection. Do not provide Windows with a Windows installation CD-ROM; instead, click Cancel in the dialog box. Windows is essentially trying to stop you from doing what you want to do, so you need to override it in this one instance.

You can find more information about this procedure at http://support.microsoft.com/default.aspx?scid=kb;EN-US;q263837, which is a Microsoft article detailing how to uninstall different versions of Outlook Express.

As always, installing a service pack or certain patches will almost certainly reinstall Outlook Express, so you need to test and be prepared to repeat this procedure if necessary.

**Windows Media Player**

Like IE and Outlook Express, Windows Media Player appears in the Add/Remove Windows Components list with a check box, implying that you can uninstall it by clearing the check box. Think it works? Of course not. Windows Media Player can actually be more difficult to remove than IE, as it’s very deeply integrated into the OS. You can typically uninstall a recently installed version (such as Windows Media Player 10), but doing so merely reverts to the previously installed version; it doesn’t completely remove all copies of Windows Media Player.

In fact, it is nearly impossible to completely remove Windows Media Player, leaving you with the alternative of making it impossible to run it. To do so, start by installing the latest-available version (from http://www.microsoft.com/windowsmedia; doing so will put Windows Media Player in a more accessible location so that you can perform this trick). Open Windows Media Player’s installation folder (typically C:\Program Files\Windows Media Player). The file you are looking for is Wmplayer.exe; edit the security settings on this file so that you can perform this trick.

As a backup plan, you might create a special domain group named Windows Media Player and allow only that group to have Full Control over Wmplayer.exe (remove all other users and groups from the file’s access control list—ACL). Simply leave the group empty and nobody will have access; should you ever need access, you can just add your user account to the group.
Although hardly an ideal solution, restricting access to the player at least keeps it from operating. A more centralized method of achieving a similar result would be to use Windows XP and Windows Server 2003 (WS2K3) Software Restriction Policies (SRP), a component of Group Policy. Create a software policy that is a hash rule of Wmplayer.exe, then set the action for that rule to be Disallowed. Figure 3.6 shows just such a rule being configured in a Group Policy Object (GPO); by applying this GPO to your computers, you will prevent Windows Media Player from running.

So why not use this same technique on IE and Outlook Express? In the case of Outlook Express, you might, but if you don’t need it, why not just get rid of it? I imagine if you had some users who needed Outlook Express and others who didn’t, and it had to be Outlook Express and not some alternative mail client, you could use the SRP or permissions technique to make Outlook Express available to some users. Its main executable—Msimn.exe—can be treated the same way Wmplayer.exe was. In the case of IE, though, this technique is insufficient. Iexplore.exe doesn’t contain much of IE’s actual functionality, and restricting access wouldn’t stop some other application (or attack) from running IE. The removal technique is much safer, if you can do it.

Figure 3.6: A rule to stop Windows Media Player from running.
**Windows Messenger**

Windows Messenger is a bit tricky. Although it is not difficult to remove, it does register itself with a half-dozen other programs, including Outlook Express. Simply deleting Windows Messenger will result in delays when those applications launch and try to get in touch with Windows Messenger.

The first step is to hide—not delete—Windows Messenger. To do so, change the registry value `HKEY_LOCAL_MACHINE\Software\Microsoft\Outlook Express\Hide Messenger` to have a value of 2. Then, execute

```
Rundll32 advpack.dll,LaunchINFSection
%windir%\inf\msmsgs.inf,BLC.Remove
```

to unregister Windows Messenger and prevent startup delays in other applications. A short VBScript can accomplish the trick easily:

```
On Error Resume Next
Set WSHShell = WScript.CreateObject("WScript.Shell")
val = "HKEY_LOCAL_MACHINE\Software\Microsoft\" & _
"Outlook Express\Hide Messenger"
setting = 2
cmd = "RunDll32 advpack.dll,LaunchINFSection" & _
" %windir%\inf\msmsgs.inf,BLC.Remove"
WSHShell.RegWrite val, setting
WshShell.Run(cmd)
```

After that is done, you might also create an SRP (or apply NTFS permissions) to restrict access to Msmsgs.exe, the Windows Messenger executable, in much the same way we did for Wmplayer.exe in the previous section.

Even more simply, you can deploy a GPO that enables the *Do not allow Windows Messenger to be run* setting. This setting works for Windows Messenger 4.0 or later on Windows XP Professional computers. Figure 3.7 shows this GPO policy enabled in a computer’s local policy.
This trick doesn’t remove Windows Messenger, but it does stop it from running. Windows Messenger doesn’t have a lot of subsidiary components that can be run independently, so stopping Windows Messenger from running will pretty much ensure that any vulnerabilities it contains can’t be exploited. Fact is, actually getting Windows Messenger completely off of a computer—as with the other bundled software—is difficult enough as to be impractical.

A final technique is to simply rename the folder C:\Program Files\Messenger; call it DisableMessenger or something, instead. Windows will be unable to locate the software, and it will be more difficult for attackers to locate it if they want to run it. Used in conjunction with the GPO method, this will cover most of your bases in ensuring Windows Messenger doesn’t allow attackers an entry point into your computers.
The Alternatives

So you’ve disabled and/or removed as much of the bundled software as you can—now what do you do? The next four sections discuss popular alternative software that comes closest to replacing the functionality of Microsoft’s bundled offerings.

Web Browser

The all-time champ of alternative software is the Mozilla Foundation’s Firefox browser (http://www.mozilla.org/firefox), which Figure 3.8 shows.

![Mozilla Firefox 1.0](image)

In addition to a host of leading-edge UI features such as tabbed browsing and a built-in pop-up blocker, Firefox supports the latest Cascading Style Sheets and HTML specifications, meaning it can display most any page that IE can. Unlike some other alternatives, Firefox isn’t just a replacement shell that instantiates IE; Firefox uses its own open-source HTML rendering engine (called Gecko).
In something approaching irony, using Firefox to browse many Web sites provides a less satisfactory experience than using IE because of the way the Web sites are programmed, not because of Firefox’s functionality. What happens is that the sites check Firefox’s *user agent*, a string the browser sends to identify itself, and determines that it can’t handle high-end HTML and styles. Thus, the sites then “downgrade” themselves to a less-robust experience. In fact, Firefox *can* handle many of these sites. A Firefox add-in called PrefBar can be used to send an IE 6.0 user agent string, fooling the Web site into giving maximum functionality, as Figure 3.9 shows. You can download PrefBar from [http://prefbar.mozdev.org](http://prefbar.mozdev.org).

**Figure 3.9: Notice the PrefBar set to give the IE 6.0 Windows XP user agent string.**

On the downside, Firefox can be more difficult to deploy simply because Mozilla inexplicably doesn’t package it in a Windows Installer (MSI) file; it’s a standalone installation executable. Of course, if you have a repackaging tool you can create an MSI from it, allowing you to deploy Firefox via GPO, if desired. A more serious downside is that, because Firefox is cross-platform, it stores its settings locally in a file, which is a technique that works for most any platform on which Firefox runs. That makes centralized management of Firefox via GPO—which affects the registry—impossible. You therefore cannot centrally configure browser proxy settings, home pages, and so forth. Hopefully these enterprise-level issues can be addressed in future version of Firefox.
A recently announced new version of Netscape Navigator (which is based on the same core code as Firefox) offers an intriguing alternative for organizations that still need to use IE. This new browser will use the same Gecko engine that Firefox does, but will also offer the option to instantiate IE right within the browser, effectively making the new Navigator both an alternative to IE and a clone of IE. This alternative would allow users to use a single browser to safely surf the Web and Web sites that require IE functionality. The browser would “remember” your engine preference on a site-by-site basis, making the process of surfing pretty much transparent. Additional details about this planned release will hopefully be forthcoming, as it seems to offer a good balance between using an alternative browser and needing functionality that is unique to IE.

Email Client
Mozilla Foundation strikes again with Thunderbird, a popular and fully functional alternative to Outlook Express. It includes the usual high-end client features, such as an address book, the ability to access newsgroups (and even RSS news feeds), and so forth. Unlike most alternative mail clients, however, Thunderbird provides full support for digital certificates, meaning you can exchange digitally signed and encrypted emails. You can find Thunderbird at http://www.mozilla.org/thunderbird, and it’s shown in Figure 3.10.
However, Thunderbird shares a number of drawbacks with Firefox, including a non-MSI installation package and no real means of centralized enterprise management. Also, Thunderbird is an alternative to Outlook Express; it isn’t an alternative to Microsoft Outlook and doesn’t contain any functionality for native connectivity to Exchange Server. As most enterprises use an enterprise messaging platform that offers its own client—such as Microsoft Exchange Server or Lotus Notes—you are more likely to use those than Outlook Express or any of its alternatives.

**Media Player**

The main competition for Windows Media Player is Apple QuickTime and RealPlayer, both shown in Figure 3.11.

*Figure 3.11: Apple QuickTime Player and RealPlayer.*
As I said earlier, if you need Windows Media format support, you’re stuck with Windows Media Player; if you don’t, you can switch. Neither QuickTime nor Real Player, however, offers Windows Installer packages (as with Firefox, however, you can make your own package if you have an MSI packaging tool) and neither offer centralized management via Group Policy. However, for a media player application, neither of these features is likely to be critical.

**Instant Messaging**

Instant messaging is the next battleground for the corporate desktop. You really have two choices when it comes to alternatives: An alternative client that still allows you to use the Microsoft instant messaging network, or a completely different network—which will come with its own client, and can be a private network if desired.

The most popular alternative client is probably Cerulean Studios’ Trillian ([http://www.ceruleanstudios.com](http://www.ceruleanstudios.com)), a client that is compatible with America Online, ICQ, Microsoft, Yahoo, and Internet Relay Chat (IRC) instant messaging networks, allowing users to participate in one or more networks simultaneously, if desired. Again, third-party software vendors tend to not accommodate Windows-based enterprises in their efforts, and Trillian isn’t available in a Windows Installer package and it doesn’t support centralized management via GPO.

---

**Why No GPO Functionality?**

I don’t know why third-party manufacturers don’t make their products more manageable via GPO because doing so is simple enough: Simply store the application’s settings in a specific area of the registry, then create text-based template files that tell the GPO Editor where those registry settings are and what they do. Doing so wouldn’t change the way the application works or prevent it from working outside an Active Directory—AD—environment, so there seems little reason not to do it. Several of these alternative software applications would be more viable if their manufacturers would package them in Windows Installer files (for easier deployment) and make them GPO-manageable (for easier management).

---

On the other end of the spectrum is a completely alternative instant messaging network, such as public networks from Yahoo or America Online, or your own private instant messaging network. The latter is becoming more popular within companies, as it provides the rapid communications benefits of instant messaging while keeping the corporate network (and its communications) more isolated from public networks. A huge variety of manufacturers offer private instant messaging solutions, including Jabber ([http://www.jabber.com](http://www.jabber.com)). Microsoft even offers Live Communications Server (LCS), a private network that uses the Windows Messenger client. Using LCS offers sufficient security benefits; by disconnecting Windows Messenger from the public network and only allowing it to use your private network, you markedly reduce the opportunities for the client to be attacked and its vulnerabilities to be exploited.
Coexistence: The Middle Ground

Although you can usually make an all-or-nothing decision with regards to Outlook Express, Windows Media Player, and Windows Messenger, IE is nearly always a sticking point. That leaves most companies in an awkward position: Ditch IE completely and risk a loss of functionality or keep using IE despite its shady security past?

There is a middle ground that might be a better choice. Use an alternative browser as users’ main choice, and use IE only where needed. For example, if users’ primary need for IE is for an intranet application, configure IE to only access the intranet, forcing users to employ a less-vulnerable and less-integrated browser in the more dangerous wilds of the Internet.

IE—especially in Windows XP SP2 and later—can be configured via GPO with a number of settings to restrict its behaviors and enhance its security (that is, make exploitable features more difficult, if not impossible, to use). Figure 3.12 shows a Windows XP Professional SP2 computer’s local policy (which is the same as the centralized Group Policy that could be used to manage this computer), and all of the IE settings which are available.

Figure 3.12: Managing IE through Group Policy configurations.
You can use third-party tools to provide even more granular configuration. For example, Figure 3.13 shows ScriptLogic Desktop Authority, which can be used to set a few dozen IE-specific settings. This figure shows a policy that removes users’ ability to change the browser’s home page; combined with other settings, you might configure IE’s home page to point directly to an intranet application that relies on IE-specific functionality. By preventing users from changing this location (hiding toolbars, disabling menu options, locking down the home page setting, and so forth), you turn IE into an application-specific tool. Users can use it only to access the one application that requires it and will be forced to use your alternative browser for other Web sites.

![Figure 3.13: Locking down IE’s functionality.](image)

If IE will be needed for a broader range of sites, you can lock down problematic functionality, such as the ability to download ActiveX controls or browser helper objects (BHOs) by configuring the appropriate Desktop Authority settings.

Other tools that can help secure IE and restrict its functionality—helping ensure that users only use it where it’s absolutely necessary—include Secure Browser by Tropical Software and IE Guardian by Devicode Technology.
Summary

This chapter has presented alternatives for the bundled software included with Windows: IE, Outlook Express, Windows Media Player, and Windows Messenger. We’ve explored how to remove, disable, or block access to these bundled applications—where possible—and covered some of the weaknesses of the alternatives. In the end, removing or disabling—where possible—Windows’ bundled applications will help reduce your computers’ attack surface; these bundled applications (particularly IE) are responsible for close to two-thirds of the Windows security vulnerabilities discovered in recent months.

The next chapter looks at securing AD, focusing—as in previous chapters—on often-overlooked areas of security, on consistency and auditing, and so forth. I’ll walk through how AD configurations become gradually less secure over time and how you can combat this security drift, and I’ll address some major weaknesses in AD’s security support features, such as reporting.
Chapter 4: Securing Active Directory

AD comes out of the box in a pretty secure state, particularly in WS2K3, which uses secure Lightweight Directory Access Protocol (LDAP) by default and uses a fairly locked-down set of default permissions and configuration settings. (Win2K also uses secure LDAP by default once you install SP3 or later.) Unfortunately, AD is literally what you make of it, meaning it comes out of the box almost entirely useless until you create users, computers, organizational units (OUs), Group Policy objects (GPOs), and so forth.

Initially, most organizations take the time to plan AD deployments and come up with a reasonably secure initial configuration. However, over time—as objects are added, removed, updated, and moved around—AD often becomes somewhat less than secure. Nobody is really to blame: AD itself is designed to give you a lot of flexibility and won’t complain if you don’t follow best practices because you may have specific organizational needs that prevent you from doing so. Administrators can’t be faulted, because too often they’re caught up in the heat of the battle, dealing with less-experienced junior administrators, inheriting environments that weren’t well-configured to begin with, and so forth. In addition, AD itself can be incredibly complex, so it’s hardly surprising that a few security practices drop through the cracks now and again.

But AD is, much like your physical network infrastructure, a core part of your organization’s operations and security. As more applications rely on AD as a central directory, making and keeping AD secure becomes more important to securing everything else in your enterprise. This chapter focuses on how AD becomes less-than-secure, how you can work with AD security more effectively, which AD security items are often overlooked or neglected, and how you can work to keep AD more secure on a continuing basis.

Becoming Less Than Secure

How AD becomes less than totally secure differs from organization to organization, but fundamentally the problem is evolution. When an organization first deploys AD, there’s often a focus—or at least a concern—on security, meaning care is taken to set up security groups, place users in the right OUs, and so forth. Many experienced administrators deploying AD have lived in older, chaotic environments and take the AD deployment as an opportunity to “do things right” for a change. The deployment becomes a revolution, meaning the old ways—and old domains—are thrown out the window and a new, done-right-this-time way is instituted. Few organizations, in other words, do an initial AD deployment that has a lot of security problems built in.
Over time, though, AD evolves to meet the organization’s continuing needs. By evolution, I’m referring to small, incremental changes made by a large number of people over a longer period of time. Users come and go, user groups are added and changed, operational and other issues drive changes to the directory structure, and so forth. Unfortunately, many of these incremental changes are made without as strong a security consideration as in the initial deployment. Often, these changes are made without a full understanding of the directory’s initial structure and the reasons behind it, meaning the changes often defeat or work around the directory’s initial security designs. For example, in the rush to fix a problem, an administrator might assign permissions in the directory directly to a user account rather than placing that user into a group that already has the necessary permissions. This direct assignment of permissions to a user account—particularly in light of AD’s complex system of permissions inheritance—often leads to unforeseen privileges being granted to that user, either immediately or later down the road as the directory evolves further.

A significant problem with AD, as it ships, is that it provides very few tools for gaining any kind of deep perspective into the directory’s security configuration. It’s difficult, for example, to determine what permissions a given user may have in various places in the directory; AD’s administrative tools simply weren’t design to provide that kind of functionality. Therein lies perhaps the biggest security risk in AD: An inability to quickly and easily see what’s already in place, security-wise. Because AD’s permissions layer on one another, making changes without seeing what’s already in place can compound existing problems as well as create new ones. So what can you do?

**Getting AD Under Control**

The first step is to get AD under control, meaning you need to design and implement a solid AD change management system. This system doesn’t need to be terribly complex, and it doesn’t need to involve expensive tools (although some great tools exist to help). The idea behind change management is that change is inherently dangerous. Why? Take an organization that has deployed a well-designed, well thought-out directory. In the beginning, everything is good—things are secure, well-configured, and so forth. It’s not until changes begin to happen—evolution—that problems begin to creep in and build up. So change is dangerous. Not bad, exactly; without changes, the directory quickly becomes useless. Change is dangerous simply because it has the potential to cause problems.

Change management attempts to mitigate those problems by putting changes through a process designed to ensure that changes are always reviewed in context of the directory’s original design. In other words, changes are reviewed to make sure they’re not defeating or bypassing the security structure that was built into the directory upon its deployment. Figure 4.1 shows a sample change management process. It’s a relatively simple process that’s designed to help safeguard the directory against changes which, over time, will reduce the directory’s overall security.
This change management process—based on best practices recommended by the Information Technology Infrastructure Library (ITIL—for more information, go to http://www.ogc.gov.uk/index.asp?id=2261)—is designed to meet several security and organizational needs. First, it begins when a change is proposed, perhaps a change to group membership or a change to permissions assignments. If the change is to group membership only, the remainder of the process is bypassed and the change is immediately implemented and documented.
Always document changes to AD. This documentation provides a central repository of knowledge against which future changes can be reviewed. If you’re simply changing a group membership, document the group’s new membership along with a reason why the change was made (“User left the company,” for example).

The theory is that AD permissions are assigned only to groups; thus, changing a group’s membership doesn’t actually directly impact the directory’s security. True, some business controls need to be placed to control who can belong to what group, but those controls would occur in advance of this process so that by the time you’re ready to make the change, it has already been approved at a business policy level.

Any other changes are reviewed by a senior administrator, who categorizes them based on their priority. Urgent changes are handled immediately by the same administrator (who must be different from the person proposing or implementing the change). The reviewer simply examines the change—which should be written in a very detailed fashion—and reviews the existing AD documentation. The purpose of the review is to catch any incompatibilities between the proposed change and the directory’s existing configuration and security design.

If the change isn’t urgent, it’s queued for a change review board, which meets on a regular basis (regular might be weekly, or even daily, depending on your organization and the volume of changes you’re dealing with). The idea is that a board composed of two or three people will catch more potential problems. For non-critical changes, waiting a few days for the board to review and approve the change isn’t a big deal, and it reserves the senior administrators’ time for urgent changes. Once approved, the change is implemented and the organization’s AD documentation updated. It cannot be stressed enough how critical this documentation is to maintaining the directory’s security over the long haul. The following list highlights examples of information that the documentation should include:

- Infrastructure information, such as where domain controllers are located, which domain controllers act as site bridges, where site links are configured, and so forth should be included.

- A listing of every group—particularly security groups, but also distribution groups—and what they are used for should be documented. In the case of security groups, a detailed list of directory permissions should be included. In the case of groups used to control file server or application permissions, document what the group provides access to.

- A listing of every user that should be included in every group. This listing can be used to audit your actual group memberships; any discrepancies are cause for investigation.

All of this documentation can be done without fancy tools. However, there are some fancy tools out there that can make change management a little easier. For example, NetPro offers ChangeManager for Active Directory, which Figure 4.2 shows. This tool provides a graphical portal through which changes can be proposed, reviewed, approved, and deployed. By enforcing your change management process’ workflow, the tool helps to ensure that the process is always followed. It also provides reporting for recent changes to various aspects of AD, such as GPOs and Sites and Services.
NetPro offers a complementary tool, ChangeAuditor for AD (see Figure 4.3), that tracks all changes that occur within AD. As part of a change management process, you must accept the fact that out-of-process changes are likely to occur. ChangeAuditor catches them as they happen and logs them to a database, often showing you not only what the change was but also what that particular setting looked like before the change was made. By providing detailed reports of recent changes and filters that allow you to focus on critical changes (such as changes to security groups), you can easily audit your environment to spot changes that occurred outside your change management process.

The ultimate goal of a change management process is to help AD remain consistently secure on an ongoing basis. Of course, if you’re in an organization that hasn’t had a change management process in place, you’ll likely have some cleaning up to do first.
Using Granular Permissions

A common problem in AD environments is permission assignments that are too broad. In fact, one of my biggest complaints about AD as it ships is that too few built-in security groups are offered, meaning everything pretty much comes down to the all-powerful Domain Admins and Enterprise Admins groups. Unfortunately, those are probably the worst groups to use on a daily basis. Using more granular permissions helps to reduce the negative effects of configuration errors, makes it easier to share permissions (and workload) throughout your organization, and so forth.

Permission Strategy

Most organizations dump the entire burden of AD administration squarely on their administrators and Help desk personnel. That’s fine in theory, but AD was designed to be much smarter and more flexible than that. For example, let’s suppose Suzie the Human Resources Director has a file server all her own on which she stores her department’s sensitive files. Company policy states that only Suzie can decide who should have access to those files. You’ve done the right thing and set up a couple of security groups in AD, and assigned file permissions on Suzie’s file server directly to those groups. However, anytime Suzie wants someone else to access the files, she has to call you to have the group membership updated. It’s silly; why not let Suzie make the change herself? Of course, there are issues around creating the exact set of permissions necessary to allow her to do that and nothing more, and of giving her a user interface through which she can manage just those two groups, but I’ll touch on those subjects later. My point for right now is that you should carefully examine how your directory is actually used, and come up with a permissions and management strategy that reflects the directory’s real-world usage.

<table>
<thead>
<tr>
<th>Principle of Least Privilege</th>
</tr>
</thead>
<tbody>
<tr>
<td>This is a good time to discuss PoLP, the Principle of Least Privilege. The idea is that any user account should only have enough permissions to do whatever that user does for most of their time. Permissions for other tasks should be assigned to a secondary user account that the user utilizes when those other less frequently performed tasks need to be taken care of.</td>
</tr>
<tr>
<td>For example, a senior network administrator in a large organization might spend most of his or her time attending meetings, working on projects, and reading email—not changing user accounts or directly administering the directory. It makes sense, then, for that administrator to use a less-privileged user account for those day-to-day tasks, and to log on (or use the Runas utility) by using a separate user account when he or she needs to administer the directory. That separate user account would be a member of the Domain Admins group, for example; the administrator’s primary account wouldn’t be.</td>
</tr>
<tr>
<td>PoLP is designed to help protect the environment against both accidents and attacks. If the administrator accidentally runs a script, for example, that modifies several accounts in the domain, the script won’t run because of the administrator’s less-privileged primary account. If a virus attacks the administrator’s computer, its effects will be limited to whatever the administrator’s less-privileged primary account can do.</td>
</tr>
<tr>
<td>PoLP doesn’t just apply to administrators, however. Suzie in HR might have a separate account that gives her permission to modify the two security groups that are used to control access to her file server’s files; you can then perform more detailed auditing of Suzie’s administrative activities by filtering for that privileged account, and you’ll ensure that Suzie’s regular account can’t be compromised to modify access to the sensitive HR files.</td>
</tr>
<tr>
<td>PoLP is something that every environment should use, although regrettably very few do, mainly because of the perceived inconvenience of having multiple user accounts per person. In fact, Windows’ built-in Runas tool, along with the ability to right-click applications and shortcuts and select a “Run as” menu option, makes using alternative credentials pretty easy.</td>
</tr>
</tbody>
</table>
A first step, then, in securing AD more effectively is to rethink how you use security permissions and user accounts. Think about more granular permissions. I like to start with the notion that nobody should be using a Domain Admin account on any kind of regular basis. That built-in group simply has such broad permissions that it isn’t safe. Instead, create your own groups that reflect the actual tasks users are performing, and assign the appropriate permissions and rights to those groups. If one of your groups starts to look suspiciously like a duplicate of the Domain Admins group, then you can go ahead and use Domain Admins for users that need all those rights. But at least you’ll have gone through the steps of validating the need for those broad permissions, rather than just accepting them because that is the easy thing to do.

Most security problems in AD come back to administrators taking the easy way out. Yes, Domain Admins is easy and when you’re a member everything works just fine. Yes, coming up with more granular permissions and groups is more difficult. But look, if you don’t want a more secure environment, why are you reading this book? Anything worth having will take some effort, and security is very worth having. Stick with me: I’ll be introducing some tools and techniques that’ll make things a bit smoother.

**The Delegation of Control Wizard**

Most administrators’ first experience with more granular AD permissions is the Delegation of Control (DoC) Wizard, built into the AD Users and Computers console. The wizard allows you to delegate specific permissions to one or more users (bad idea) or groups (best practice), for a specific object or objects within AD. In other words, you might give the HRAdmin group permission to reset passwords on user objects within the HR OU. As Figure 4.4 shows, the wizard supports several preprogrammed permission sets, such as resetting passwords, managing Group Policy links, and so forth, or you can create your own custom tasks (permission sets).
The only thing I don’t like about the wizard, especially when using one of the preprogrammed permission sets, is that it’s doing stuff behind the scenes. That’s its job, really—to perform the permission change while hiding some of the underlying complexity. Until you fully understand that complexity, however, I think it’s more valuable to make the permission changes yourself. That way you’ll know what’s happening under the hood, and you’ll understand the security consequences of what you’re doing more clearly.

For example, as Figure 4.5 illustrates, a single run of the wizard creates a fairly complex underlying set of permissions, requiring you to navigate through a series of dialog boxes to see what changed. The wizard didn’t do anything wrong or inaccurate, but it might benefit me to know that, for example, the permission change the wizard made isn’t inherited to sub-OUs; the permissions I delegated only apply to the OU I selected when running the wizard. The permission applies only to User objects, something else that the wizard glosses over a bit.
Figure 4.5: Actual permissions set after the DoC Wizard runs.

The wizard is more valuable for administrators who know how to perform its job manually, understand what the Wizard is doing under the hood, and simply use it as a timesaving device. I don’t like newer or less-experienced administrators to use the wizard simply because it allows them to make significant changes with a much less thorough understanding of the result. I prefer that they manually work with AD permissions first so that they become more familiar with what’s going on under the hood.

**Manually Working with Permissions**

AD permissions are complex because they’re so flexible (or maybe it’s the other way around). AD permissions allow you to:

- Give permissions to a group or user
- Assign permissions to a specific object, such as a user or OU
- Assign permissions to children of a container or OU (for example, you might assign permissions only to users within an OU but not the OU itself)
- Allow permissions to inherit to child objects, such as sub-OUs
Right-clicking an AD object (in the AD Users and Computers console, for example) and selecting Properties provides access to the Security tab (see Figure 4.6). Unfortunately, AD permissions are far too complex to be easily represented by this dialog box, so you’ll nearly always find yourself facing grayed-out check boxes (indicating permissions that are inherited rather than being assigned directly to the object in question), as well as the Special Permissions check box.

Figure 4.6: Basic AD permissions.

Dealing with Special Permissions—which nearly always exist—requires you to use the Advanced Permissions dialog box, which Figure 4.7 shows. Even then, you’ll often find yourself confronted with “Special” permissions, requiring you to click the Edit button and face a third dialog box.
The Advanced Permissions dialog box (see Figure 4.7) does show some important information. It indicates where inherited permissions came from (or indicates that a permission wasn’t inherited at all, but was assigned directly), and it indicates the scope of the permission: this object only, this object and all child objects, a specific type of object (such as User or Group), and so forth. You can also use the *Allow inheritable permissions from the parent to propagate to this object and all child objects* check box. When selected (the default), all permissions defined on the parent as inheritable will be inherited by this object; when cleared, no permissions will be inherited from the parent—even if said permissions were intended to be inherited. Clearing the check box effectively blocks permission inheritance, allowing you to start with a clean security slate.

This manual application of permissions lets you get very granular. You could, for example, allow members of a specific group to modify the membership of any other groups located in a particular OU, without giving them that ability in other OUs, any sub-OUs, and without giving them the ability to create any new group (or other) objects, and without giving them any control over groups in any parent OUs. It’s flexible…and complicated.
Ultimately, however, coming up with complicated, granular permissions is the way to go. You’ll distribute authority throughout your organization in a way that reflects the actual control in your organization. To use my earlier example, Suzie is the one with authority over who can be in those two HR-related security groups; AD permissions should reflect that authority whenever possible rather than forcing Suzie to call you. Since you don’t have business-level authority to modify those groups, you shouldn’t necessarily have AD-level permissions to do so, either.

<table>
<thead>
<tr>
<th>Distributing Authority</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delegating control is often ignored because organizations believe it’s too difficult to give users a user interface appropriate to their level of permission. For example, giving Suzie permission to modify her two security groups is easy; giving her a user interface to do so is more difficult. You could just give her AD Users and Computers; she wouldn’t be able to do anything except what she’s allowed to do. However, it’s a poor security practice to expose people to things they have no permissions to use, so the entire AD Users and Computers console would be overkill.</td>
</tr>
<tr>
<td>One alternative is to use the Microsoft Management Console (MMC) to create a TaskPad view of the AD Users and Computers console, creating a subset of the console’s functionality that reflects only what Suzie can do. Another alternative would be to write scripts or develop small custom applications that just have enough functionality to do what a particular user needs.</td>
</tr>
<tr>
<td>Companies such as NetIQ offer products designed to address this need. Its Security Administration Suite, for example, provides an in-product system of delegation and a customizable user interface that provides delegated individuals with just the user interface elements necessary to do what their permissions allow.</td>
</tr>
<tr>
<td>Whatever you decide, the point is that you can create limited user interfaces to reflect delegates’ limited permissions and capabilities. Don’t let that be a barrier in creating a more delegated security system in AD, one that more closely reflects that actual business authority in your organization.</td>
</tr>
</tbody>
</table>

Maintaining Consistency

One step to making AD more secure—and a major reason to adopt a good AD change management process—is consistency. Consistent configurations are more reliable simply because they’re the same. For example, you may need to delegate password-reset permissions to multiple groups of people. Doing so the same way, every time, will help ensure that the delegation is done correctly. Consistency generally makes everything easier in information technology, and AD security management is no exception.

Command-Line Utilities

One way to help maintain consistency is to use scripts or command-line utilities. By performing delegation tasks, for example, using a utility or script, you’ll be assured that the task is done the same way every time you do it. You don’t need to worry about skipping a step in a manual process, forgetting to check a critical check box, or leaving out some other step; the script or utility will always do things consistently.

Microsoft provides a command-line utility named Directory Services Access Control Lists (DSACLs.exe), which is designed to make changes to AD permissions. The tool can be added to a batch file to make sure the file is always run consistently for various operations; you might have one batch file for each common permission- or delegation-related task that you perform. DSACLs is also a great way to apply batch corrections to incorrect security permissions within AD, as it can affect numerous objects at once, much more quickly than you could do manually.
An example of the tool’s syntax:

```
Dsacls "OU=Peeps,DC=company,DC=com"
    /G Company\JohnD:CCDC;group;
```

This text would give user JohnD permissions over the OU Peeps. The permissions granted are Create Child and Delete Child specifically for groups; in other words, JohnD can create and delete groups within the Peeps OU.

There’s a downside to using tools like this, however. First, the syntax is anything but straightforward, and you’re as likely to make a mistake running DSACLs as you are doing these tasks manually. Other command-line tools exist, but the bottom line is simply this: AD security permissions are complex, and any tools used to work with those permissions are going to be equally complex.

**Template-Based Administration**

Template-based administration is becoming more popular in the world of IT management, especially when it comes to security. The basic idea is that you create a bunch of templates (or rules, or policies, or whatever you want to call them) that describe how various aspects of your IT environment should be configured. You then use tools to actually apply those template settings to your environment, automatically. Oftentimes, the tools have the ability to continually enforce your templates, overwriting any unauthorized changes that are made.

An example of a product that uses this technique is ConfigureSoft Enterprise Configuration Manager (ECM). With ECM, you might define a template that requires the Messenger service to be stopped and disabled. You apply the template, using ECM, to whatever machines should be governed by that template. ECM inventories the machines and corrects their configurations to match the template. If a user or administrator manually enables the Messenger service, ECM detects that and reconfigures the machine—automatically—to comply with the template. If you ever want to enable the Messenger service, you just change your template, and ECM reconfigures everything to match.

ScriptLogic Active Administrator provides this kind of template-based administration for AD. As Figure 4.8 shows, you can create templates that assign specific permissions. In this example, I’ve opened the permissions on the West/Sales OU in a domain, and delegated the *Groups-Modify the membership of groups* security template to the SecAdmins user group. That group will now have the template’s permissions on that particular OU.
This capability isn’t remarkably different from AD’s DoC Wizard, except that with this method you can define your own templates anytime you like. As Figure 4.9 illustrates, you have a wide range of templates to start with, and modifying them or creating your own is easy. This method allows you to create templates for whatever AD tasks are performed in your organization, making it easier to delegate permissions for those tasks to whatever users and groups require them.
Figure 4.9: Examining a security template.

Where Active Administrator really shines, though, is when you later change a template, as Figure 4.10 shows. In this example, I’ve modified the template I previously delegated to the SecAdmins group. I’ve added the ability to create new group objects. When I save the template, Active Administrator notes that it is already applied to one OU, and that saving the modified template will affect the permissions on that OU. In other words, simply by changing the template, I’m affecting AD permissions wherever this template was used.
This solution is a powerful, template-based means of managing AD security. You might create templates that reflect the capabilities certain groups or users should have within your environment; as those capabilities change, you simply modify the templates and permissions are updated accordingly. You can begin to move away from direct (or native) AD permissions assignment and begin to work entirely with template-based permissions, improving the efficiency of your overall administration and making delegation of permissions much easier.
Security Reporting

In addition to maintaining consistency, security reporting is a key factor in secure AD administration. As I previously mentioned, AD permissions are too complex to easily view in a graphical user interface (GUI). To address this problem, you can again turn to a third-party tool that offers a clear interface to view the native permissions on any AD object, as Figure 4.11 shows.

![Figure 4.11: Viewing permissions in Active Administrator.](image)

As this figure shows, inherited permissions are tagged with an (I) indicator and listed in gray; permissions that are the result of a template application are shown in bright blue. This screen makes it much easier to view and report on AD permissions because the permissions are displayed in a very effective manner. Figure 4.12 shows that template applications can be viewed separately, allowing you to quickly note which security templates are in effect on the AD object you’re examining.
Another valuable reporting tool is NetPro SecurityManager. It can be used to get a quick snapshot of your security situation (see Figure 4.13). The domain examined in the screen shot has 413 non-conformant issues, with those variances broken down per system in the Variances Summary section. This tool, preloaded with several best-practice configuration settings, allows you to quickly spot potential security issues and deal with them appropriately.
The purpose of any AD reporting tool is to simply help aggregate and filter the immense amount of security information available under AD. Whether the tool is consolidating and filtering AD auditing events, AD security settings, or a broader range of AD configuration settings, reporting allows you to view, in one place, a larger amount of information than AD’s native tools allow you to do. Other useful AD reporting tools include:

- **Imanami SmartR**—This flexible tool allows you to produce a variety of custom reports. You can quickly find users based on their last login date, see who owns particular groups, and so forth.

- **Javelina Software AD Toolkit**—This tool provides several built-in reporting capabilities, many of which are useful for cleaning up security problems (such as inactive computer accounts and disabled accounts), as well as offers group membership reports and more.

- **Quest Enterprise Directory Reporter**—This tool is a comprehensive reporting solution intended to help you determine which users have administrative rights, what accounts are unused, how many users have been created in the past month, and so on.

- **ScriptLogic Active Administrator**—In addition to providing configuration and maintenance capabilities, this tool provides reporting on your existing AD configuration.

These and other reporting tools can help you gain better insight into your AD security situation, spot potential problems, and more easily find areas that need your attention.
Overlooked (and Ignored) Security Problems

Perhaps one of the best things you can do for AD, security-wise, is keeping up-to-date on Windows service packs and patches. That’s hardly news—patch management is a hot topic these days. But most organizations also have some easily overlooked and ignored security problems that can be a lot more dangerous than being behind on your patches.

Unnecessary Security Principles

Any security account—whether it’s a user or a computer—shouldn’t exist unless it’s being actively used for a defined purpose. Old user accounts should be disabled for a fixed period of time and deleted as soon as is practicable. Any security principle (such as a user or computer) can potentially be used as an attack point. Pretty much by definition, unused accounts’ passwords aren’t changed regularly, making them somewhat more susceptible to attack.

Built-in Accounts and Groups

I recently attended a lecture on the use of built-in accounts and groups, where the speaker proposed that built-in accounts and groups should never be used and should in fact have all of their built-in permissions removed. The argument was an extension of the common practice of renaming the built-in Administrator account. The theory there is that the Administrator (and Guest, for that matter) account name is well-known and is the first point of attack because an attacker only has to guess the password. The speaker’s argument was that the security identifiers (SIDs) for all built-in accounts and groups are well-known, and changing the account name doesn’t change the SID. Because the SID is well-known (or, in the case of a domain, can be guessed relatively easily), the accounts should be disabled, the groups should be emptied, and both should have their built-in permissions and rights removed. You should create your own accounts that have any permissions and rights you need because the SIDs of those accounts will be difficult for an attacker to predetermine.

I haven’t worked with any organizations that are implementing the speaker’s suggestion, but it’s an interesting theory. Although “security through obscurity” is no security at all, making it difficult for an attacker to know or learn anything about your environment is definitely part of a good overall security strategy.

Many reporting tools can help spot unused accounts. Windows itself keeps a “last logged in” date for all domain accounts, making it relatively easy to determine which accounts aren’t in use. Be warned, however, that prior to WS2K3, AD didn’t replicate that “last logged in” account attribute. Thus, a reporting tool must check with every domain controller to find the most recent “last logged in” date. In an all-WS2K3 domain, however (which is operating at the highest domain functional level), that attribute is replicated, so you can get an accurate last login date for any account from any one domain controller.
**Disabled Accounts**

Like unused accounts, disabled accounts represent an unnecessary security risk. Accounts can too easily be re-enabled, and it’s likely they’ll have an old password and obviously aren’t being regularly used or checked. Disabled accounts are fine for a short period of time while, for example, you take ownership of the former user’s files and other resources, but disabled accounts should be scheduled for deletion as soon as possible.

In this case, any WS2K3 domain controller (even if your domain is not entirely WS2K3) can provide you with a list of disabled accounts. Simply use the “Saved Queries” functionality of the WS2K3 AD Users and Computers console. As Figure 4.14 shows, you can easily construct a query that shows all disabled user accounts, allowing you to review whether those accounts can be deleted.

![Figure 4.14: Viewing disabled user accounts.](image)

- The Saved Queries function can also provide you with “last logged in” dates, but will only be accurate in an all-WS2K3 domain.
If a disabled account must remain in the domain, be sure to periodically change its password to something long and complex (a pass *phrase*, rather than a pass *word*, for example). Doing so will help prevent the account from being compromised in the event it becomes re-enabled. You should also document accounts that must remain in the domain and will be disabled so that you can periodically check them to ensure they *remain* disabled.

Don’t forget about computer accounts, which can also be disabled and which represent as much a security risk as user accounts.

**Accounts with Non-Expanding or Old Passwords**

AD Users and Computers’ Saved Queries function can also help find accounts with non-expiring passwords. You can use it to quickly define a query that spots non-expiring passwords, which are a major security risk (see Figure 4.15).

Most non-expiring passwords are for service accounts. Chapter 6 will address ways in which service accounts can be made more secure, including making it easier to regularly change their passwords.

![Figure 4.15: Querying for non-expiring user accounts.](image)
To query for old passwords, you’ll probably need to use a third-party reporting tool, such as one of the ones mentioned earlier. If you have a maximum password age set in your domain, locating accounts with passwords older than the maximum age is a good way to spot accounts that aren’t being regularly used. Contact the person supposedly using the account, if possible, and find out why the account isn’t being used or why the password hasn’t changed.

Don’t forget about computer accounts! While computer account passwords are automatically changed by the computer owning the account, finding accounts with non-expiring passwords—something an administrator can configure—will help spot accounts that could potentially be a security risk.

Unlinked GPOs

Another potential security problem—more from the a “security of your operations” standpoint than a “loss of data” standpoint—is unlinked GPOs in the domain. These don’t often occur, but when you unlink a GPO from the last container to which it was linked, you can leave the underlying GPO in place. The result is an “orphan” GPO, one which sits on every domain controller and contains configuration settings that could potentially disrupt your environment were someone to relink or mislink it. Although this occurrence isn’t common, an overriding part of a good security philosophy is to not have anything lying around unless it’s actually being used, so you should get rid of orphaned GPOs.

Fortunately, ScriptLogic Active Administrator makes it easy to spot them. As Figure 4.16 shows, the tool can quickly locate any GPOs that exist but aren’t being utilized; in this example, the SecondaryUsers GPO is highlighted in red, indicating that it isn’t in use.
Summary

AD comes out of the box configured as a fairly secure system. Most companies do the right thing on their initial deployment and have a pretty secure initial state. Over time, however, the sheer complexity and flexibility of AD sets in and can easily create a less-than-ideal security situation. The key is to remember a few easily overlooked security risks, such as poorly understood permissions inheritance, unused security principles, and so forth. Another key is to use third-party tools to supplement AD’s lackluster built-in security tools. The third-party tools introduced in this chapter can make permissions management easier, highlight security problems you may not have known existed, and provide you with more readable and usable reports on the status of your directory’s security.

The next chapter will step away from AD a bit and talk about file server security. File servers are probably the most abundant type of Windows server in any environment, and they have several easily overlooked security problems that, if left uncorrected, can undermine your security efforts across the enterprise.
Chapter 5: Securing File Servers

A file server is one of the most common roles for a Windows server. After all, Windows got its start in business as a departmental file server, and file services is still one of Windows’ biggest strong points. Some organizations have had file servers in place for years, so it’s no surprise that security on these servers isn’t at the highest possible level. This chapter will explore the most common security problems with file servers, and show you ways to easily address those problems, resulting in a more secure Windows enterprise.

When Good File Servers Go Bad

As with most security issues, time is the file server’s enemy. Most administrators who are setting up a new file server do so with the best of intentions, following every security best practice they know; over time, however, those best practices may be followed less rigorously, or may even change, resulting in a file server with less-than-stellar security. Figure 5.1 shows every administrator’s worst nightmare (or what should be their worst nightmare): file and folder permissions assigned directly to one or more users rather than a group.

![Figure 5.1: File permissions assigned to a user rather than a group.](image-url)
Per-user permissions are perhaps the most common security problem on file servers. Per-user permissions are more difficult to manage than group-based security settings and become increasingly complex over time as users leave the company, move to new roles within the company, and so forth. Although the best practice is to assign permissions only to groups and to place users in the appropriate groups in order to give them permissions (other than on users’ home directories, where per-user permissions are the norm), the scenario that Figure 5.1 shows is far from uncommon in most environments. Per-user permissions creep in over time as harried (and sometimes less-knowledgeable) administrators do whatever is necessary to get their jobs done as quickly as possible.

Later in this chapter, I’ll give you some tips for solving the per-user permission nightmare.

Additional security problems also exist on most file servers. For example, in most Windows environments, users can see every available file share and folder, even if they don’t have access to those resources. Windows makes this poor practice difficult to avoid. File servers are often configured with less-than-ideal security settings at the network level, too, opening the environment to spoof attacks (in which users access a rogue server that they think is legitimate), poor authentication protocols, and more. In addition, file servers may not be well-configured for data resilience, which can be a security problem when critical data is lost or intentionally destroyed or altered. Even well-meaning administrators, seeking to remove old files from expensive storage resources, can create security and operational issues by inadvertently removing critical corporate data from a file server. All of these security problems are, however, easily corrected if you have access to the right techniques and technologies.

File servers are, as their name states, servers. There are a lot of security issues that pertain to servers in general, rather than file servers in particular: Service configuration, password policies, and port management are just a few of these issues. Chapter 6 will address these more generic server security problems; this chapter will focus on the issues unique to file servers.

What You Can’t See Won’t Hurt You

If you’ve been working with networks for a while, you probably know that Novell NetWare—all the way back in its 2.x versions—had a useful feature that prevented users from seeing network shares to which they didn’t have access. This feature was useful from a usability standpoint because users wouldn’t be calling the Help desk asking why they couldn’t access a particular share, when in fact they didn’t have permission to do so. From a security standpoint, it was a useful feature because it helped prevent users from trying to work around security settings on items to which they didn’t have access simply because the users wouldn’t realize that those things were there to begin with.
It’s odd that Microsoft never implemented a similar capability in Windows, especially because Windows started out competing primarily with NetWare. Finally, WS2K3 Service Pack 1 (SP1) includes a feature called Access-based Enumeration (ABE). Essentially, ABE prevents the server from showing any resources to which a user doesn’t have access, including shares and the folders living beneath a share. Unfortunately, ABE isn’t intuitive to configure. It is turned off by default and must be turned on for each individual share. Further, there is no graphical user interface (GUI) from which to configure ABE. A free third-party utility called Sheflgs.exe (available from JoeWare.net) can be used to simplify the process. Run

```
Sheflgs \Server\Share /abe true /forreal
```

to enable ABE on \Server\Share. Sadly, you still have to do this on a per-share basis, and there is no way to extend the functionality to WS2K3 machines that are not running SP1 (nor can you use ABE on Win2K servers).

My preference is to turn on ABE for every share on a server. There is very little reason not to—if a user doesn’t have access to a share, there is no reason for the user to see it, right? Unfortunately, the fact that ABE only exists in WS2K3 SP1 and that it has to be configured on a per-share basis is certainly a bit limiting (that there is no graphical means of activating it and it isn’t turned on by default is a bit bewildering).

Another option—one that includes a GUI and runs on Win2K servers—is to use a third-party tool. As Figure 5.2 shows, a third-party option is easy to operate, and can be configured on a per-volume basis to hide files and folders from either local or network users.

![Cloak Manager NOT FOR RESALE](image)

**Figure 5.2: Configuring ScriptLogic Cloak to hide access to files and folders.**

Cloak’s ability to hide local files and folders means it’s suitable for use on shared client computers, allowing you to hide local files and folders to which the currently logged-on user doesn’t have access.

Remember, these tools aren’t hiding items to which users have access; it’s only hiding resources that the user won’t be able to access anyway.
Note that this approach is very different than simply adding a dollar sign ($) to the end of the share name. That technique hides the share from the browse list of all users, whether they have access or not; I dislike that technique if it’s being used to hide shares from users who have access simply so that those users won’t realize they have access. If you don’t want someone accessing a share, don’t give them permission to do so—don’t rely on obfuscation (for example, hiding the share) to protect the share’s contents. Third-party tools and ABE simply take their cue from the underlying file, folder, and share permissions, permitting users to see only what they already have rights to access.

One area in which either ABE or a third-party tool can be useful is in user home directories. With ABE or another solution in use, you can map all users to a generic share, such as \Server\Users, under which you’ve created individual folders for each user. With the appropriate permissions applied to each subfolder, ABE or the third-party tool will ensure that users see only their own subfolders and not other users’ subfolders. This setup removes the need to create individual shares for each user, making network management much more convenient and efficient.

Of course, a key to all of this hide-the-share capability is having the correct file, folder, and share permissions in place to begin with—a setup that we’ll explore toward the end of this chapter.

At the Network Level

File servers—especially those that have been in the environment for a while and may have been upgraded from older versions of Windows—often have the most security issues at a network level. Older versions of Windows lack the security features of newer versions, and as older file servers are upgraded, administrators often overlook the need to investigate and configure new security options. Many of these options are, however, targeted at making file services more secure, and they’re worth looking into.

Although this chapter focuses on WS2K3 as the server operating system (OS), the majority of these features and settings are available, in some fashion, on Win2K as well.

Figure 5.3 shows some of the security policies that can be configured to make file services more secure in general. Note that some of these policies also apply to the client computers in your environment—because they relate specifically to file services (for example, client interaction with file servers), they’re covered here.
The settings shown in the figure are the default settings for a newly-installed WS2K3 server; the settings active on any particular server will be different if it was upgraded from an older version of Windows. Also note that the figure is displaying these settings from the computer’s Local Policy; these settings can be configured on a per-server basis in Local Policy or can be centrally configured in AD using Group Policy (which is easier, more consistent, and definitely the recommended means of doing so). Key settings include:

- **Microsoft network client [and server]: Digitally sign communications.** There are two settings here; one that always signs communications and one that only does so if the server agrees to do the same. This policy is useful to apply on clients and servers because it allows computers to detect rogue servers that are attempting to fake (or spoof) a network connection in order to intercept information. Enable these settings for the best security.

  Enabling signatures can create problems for older client computers (such as Windows 98) that don’t support this feature; investigate the possible ramifications of enabling this setting in your environment before doing so.
• Do not allow anonymous enumeration of the Security Accounts Manager (SAM) accounts and shares. The *shares* part of this setting is critical—allowing anonymous users to list the shares available on a server is a bad idea because it helps an anonymous attacker get an idea of which resources are available to attack. Enable this setting for better security.

• Let Everyone permissions apply to anonymous users. For best security, ensure that this setting is disabled, preventing the built-in “Everyone” group from including anonymous, unauthenticated users.

• Restrict anonymous access to Named Pipes and Shares. Once again, anonymous equals attacker in a security expert’s point of view, so enabling this setting will help defeat anonymous attackers seeking to gain information and access.

• Shares that can be accessed anonymously. There are a couple of shares in a domain environment that are typically meant to be accessed anonymously; this setting allows you to list them (the default settings are COMCFG and DFS$, required for COM configuration and Distributed File System). Ensure that no unnecessary shares are listed.

Another area of network security that file servers can benefit from is port restrictions. Normally, I’m a big fan of locking down servers’ ports so that only the ports absolutely necessary to the servers’ operations are open. However, I’m absolutely clear on how terrifying it can be to mess with a server’s ports: One wrong move and the whole company is screaming at you. File servers in particular are scary because they use so *many* ports. Despite numerous Microsoft articles listing required ports and so forth, most administrators are quite understandably reluctant to touch their servers’ port configurations.

To ease fears, you can use WS2K3 SP1’s Security Configuration Wizard (SCW). You have to install the SCW, which is an optional Windows component available to you after SP1 is installed. SCW can be used to generate templates, which can in turn be used to configure one or more servers. The beauty of the SCW is that, under the hood, it uses a Microsoft-produced XML file that describes various roles (such as file server or domain controller) that a server can fulfill. For each role, Microsoft has determined which services, ports, and other resources are required. In other words, you tell SCW that your server is a file server and it takes care of the rest, locking down unnecessary ports, services, and so forth.

---

> You don’t need to install the SCW on every server, or even run it on every server. When you run the SCW, it creates a security template; that template contains the configuration information you need. Templates can be applied manually by using the command-line Secedit.exe tool or imported into a Group Policy Object (GPO) and applied through AD.

---

Figure 5.4 illustrates the SCW’s Viewer, which shows the configuration database after SCW tries to figure out which roles your server is already fulfilling based upon the services and applications already running. In this example, the server is a file server but not a domain controller.
Figure 5.4: Using the SCW Viewer to examine configured server roles.

After analyzing your server, SCW will ask you—as Figure 5.5 shows—which roles you want the server to fill. As shown, it starts by assuming the server is in fact meant to be running all of the software SCW detected. If, for example, you didn’t mean for this server to be running IIS, you would clear the Application server check box, the Web server check box, and so forth. The check boxes shown can be filtered to include only roles for which software is already installed (as shown), all possible roles, and so forth.
Figure 5.5: Configuring the intended roles.

Remember, changing check box settings won’t automatically reconfigure your server and break anything. The SCW’s end product is a template, or policy, not an immediate reconfiguration of the machine.

As Figure 5.6 illustrates, the SCW also looks specifically at the TCP/IP ports that are open on the computer, pre-selecting the ports that are required for the roles you selected for the server. This functionality is the SCW’s real strength because its underlying XML configuration file knows which ports go with which roles and features of Windows, eliminating guesswork on your part.
Figure 5.6: Configuring allowed TCP/IP ports.

Figure 5.7 shows another file server-specific option of the SCW. The wizard prompts you to find out whether the server has sufficient extra processor capacity that can be devoted to digitally signing file and print traffic. By indicating that the server does, SCW will configure the resulting policy template to enable Server Message Blocks (SMB) signing, a feature that can help eliminate spoofing and other forms of attack on your network.
Why bother locking down ports and services to begin with? Because Windows has bugs. With millions of lines of code, the OS will probably always have bugs. By disabling unused ports, you can prevent anything from accessing portions of the OS—primarily services—that aren’t in use and potentially may contain bugs. By restricting the services that are running (a concept I’ll cover in more detail in the next chapter), you prevent code from running which isn’t necessary and which might contain bugs. The SCW is, in fact, a tacit acknowledgement on Microsoft’s part that Windows probably contains undiscovered bugs, that not every feature of Windows is needed on every server, and that disabling and preventing access to unused features will help mitigate the fact that they probably contain undiscovered vulnerabilities.

*Figure 5.7: SCW options that enable file and print traffic signing.*

The SCW has applications beyond file servers, so the next chapter will examine it in more detail.
Data Resilience

Data resilience—the ability for an administrator or a server to deal with data loss—is often seen as a purely operational issue. However, lost or corrupted data can also be a security issue. For example, if an attacker is able to gain access to network configuration documentation and modify it, the attacker might be able to convince an administrator to reconfigure the network according to the modified documentation (convincing someone to do something they shouldn’t is a technique often referred to as social engineering, playing off the human weak point in the network’s security). That’s just one example; there are a number of ways in which illegally modified or deleted data can result in a security problem. Thus, data resilience has a clear security application as well as the more traditional, operational application.

Volume Shadow Copy

Other than the time-honored practice of making frequent backups of your data, Windows includes some nice features to help improve data resiliency and make data recovery easier and more convenient. One of those features is Windows’ Volume Shadow Copy service capabilities.

As Figure 5.8 shows, VCS is enabled on a per-volume basis. Once enabled, all files located within a shared folder will be shadowed, meaning Windows will retain old versions of files for faster recovery. These old versions are retained in a shadow copy area, which can be located on the same volume or on a separate volume (which is more appropriate for high-volume file servers). Windows doesn’t create a shadow copy for every change made to a file; instead, snapshots are taken on a schedule that you can set (the default is two per day).

Hidden administrative shares—such as the default C$ share—aren’t counted by the Shadow Copy panel, as Figure 5.8 illustrates. In the figure, the number of shares is zero, even though we know there’s a default C$ share present.
Some recommendations regarding the Volume Shadow Copy service:

- Enable it on all file servers, but only on the volumes that contain user share folders (for example, those shares that contain user-accessible files).

- Set a size limit for the shadow copy storage area based on how many files tend to change per day and how frequently you’re taking shadow copy snapshots. When the storage area is full, older shadow copies are deleted, so setting the storage area too small will result in too few copies being retained.

During a Volume Shadow Copy snapshot, only files that have changed since the last snapshot are backed up into the shadow copy storage area, so the rate at which that storage area fills up depends on how frequently files change on the server.

- Don’t schedule more than one snapshot per hour, and schedule fewer for especially busy file servers.

- Configure the shadow copy storage area for a separate volume. Ideally, this volume might be a non-RAID volume (because the data is just backup data, anyway) dedicated to the Volume Shadow Copy storage.
Clients will need to install a client component—which comes on the Windows Server CD-ROM—in order to access shadow copies. This client, called the Previous Versions Client, allows Windows Explorer to access the inventory of shadow copies for any given file, and allows users to retrieve past versions of a file (which can be saved to a different location, to avoid overwriting the current version of the file). You can reserve this functionality for administrators by simply not installing the client software on your users’ client computers.

The Volume Shadow Copy service performs essentially the same task as a traditional disk- or tape-based backup but makes the backed-up files more readily accessible. Microsoft’s intention with the Volume Shadow Copy service was to make it possible for users to retrieve recent versions of files without having to contact an administrator; this same functionality, however, has a useful security application because it can be used to protect sensitive files and quickly retrieve older versions in case an unauthorized change (or deletion) occurs.

**Storage Management, Hierarchical Storage, and Antivirus**

WS2K3 (and Win2K, for that matter) includes Remote Storage services, a basic hierarchical storage management system. You can read more about it at [http://support.microsoft.com/?kbid=816588](http://support.microsoft.com/?kbid=816588); essentially its job is to monitor the free space available on a server’s local drives. When the free space drops below a designated point, Remote Storage locates infrequently used files and moves them to an attached tape or optical drive. The moved files retain a pointer in the main NTFS file system, meaning that users can continue to see the files when browsing the network. When a user attempts to access a moved file, Remote Storage locates it on tape (or optical disk, or whatever) and moves it back into the main file system of the server. There is a delay while the file is accessed, but the process is automatic.

Remote Storage is only available on the Advanced (or Enterprise) and Datacenter editions of Win2K and WS2K3.

This process is called hierarchical storage management (HSM) because it creates a hierarchy of storage capabilities: *Online* storage (local disks), which are fast but generally more expensive; *near-line* storage (such as tapes) which are slower but cheaper; and even *offline* storage, where data is moved off the server completely and an administrator is required to retrieve it when needed. Third-party vendors such as Legato and VERITAS offer more robust HSM solutions, helping to further automate and streamline the process of moving data between online, near-line, and offline storage systems. Figure 5.9 illustrates the various levels of storage usually included in a full HSM system.
So what does HSM have to do with security? HSM helps security in a number of ways:

- Because HSM can migrate data off expensive file server storage, data doesn’t need to be deleted, therefore reducing the risk that critical data will be deleted by accident.
- HSM can also help prevent file servers from becoming filled and unavailable, because older data is moved offline or near-line, ensuring that the file server can continue to function.
- Because third-party HSM solutions are often integrated as part of an overall backup and recovery package, disaster recovery becomes an easier, more automatic operation, helping to guard against the loss of critical data.

Although the security implications might not be obvious, storage management in general—including HSM as well as traditional backup and restore—is really all about security—the security (that is, the safety) of your data. Other forms of storage management can play an important role in securing file servers, too. For example, VERITAS Storage Exec can help ensure that unwanted files stay off file servers, which helps to guard against operationally problematic files (such as copyrighted MP3 files or inappropriate graphic files), as well as outright dangerous files (such as file types known to be associated with viral attack vectors, such as VBS files).
Finally, one last and perhaps more obvious storage management category is antivirus software, which plays a crucial role in helping to secure file servers. Virus scanning should occur at every possible level of your enterprise. Client computers may all have antivirus software, but it is difficult to ensure that the software is always kept up-to-date. By adding antivirus scanning to each file server, you can help ensure that viruses don’t propagate through the file servers’ centralized storage resources. Most major antivirus manufacturers offer solutions specifically for file servers: Panda Software offers FileSecure, Trend Micro offers ServerProtect Antivirus, McAfee offers VirusScan Enterprise, and so forth. Most of the enterprise-level solutions are more easily managed from a central console than per-client antivirus software. For example, Figure 5.10 shows Panda Software’s AdminSecure console, which provides management of all server-based antivirus software including Exchange servers, Novell file servers, Windows file servers, and even Windows workstations.

![Figure 5.10: Panda antivirus software console.](image)

Of course, server-based anti-spyware scanning—available from most of the same third-party vendors—is another critical piece of security for file servers, as spyware can not only damage data but also help deliver it off your network to unauthorized individuals.

One industry best practice is to use different manufacturers’ virus and spyware scanning solutions at different levels. For example, if you’ve adopted Symantec (Norton) software on your client computers, select a different vendor for server-based scanning, a different one for firewall-based scanning, and so forth. The theory is that no one solution catches everything, so by using a variety, you’re more likely to have fewer holes through which malicious software can slip.

Thus, while storage management—including Volume Shadow Copy, antivirus scanning, anti-spyware scanning, and HSM—plays a sometimes less-than-obvious role in file server security, it is nonetheless an important role. Proper storage management can help keep critical data safe, undamaged, unmodified, and uninfected, all of which helps contribute to a more secure Windows-based enterprise.
Applying Proper File Permissions

I’ve saved the biggest and best for last—file permissions. Perhaps the most obvious portion of file server security, file permissions are also one of the most complex. Although Windows offers exceedingly complex and granular file permissions capabilities, complete with permissions inheritance, it also offers a relatively poor GUI through which these complex permissions are managed. For example, Figure 5.11 shows the basic Windows file and folder permissions dialog box. Notice that the check boxes, used to assign permissions are unavailable (grayed out). This element visually indicates that the permissions have been inherited from a parent folder, and that they therefore can’t be changed directly on this folder without first disabling inheritance and copying the existing inherited permissions directly onto the folder. A simple, grayed-out check box carries a lot of hidden, less-than-intuitive meaning!

Even worse, many permissions are special, meaning they’re not fully displayed in this simplified dialog box. Figure 5.12 shows the advanced permissions dialog box, which lists in greater detail the permissions applied to this folder.
This screen helps you determine where inherited permissions are coming from (in the Inherited From column) as well as enables you to control the more detailed and granular permissions that can be associated with this particular object. Auditing—a key portion of any secure enterprise, especially those dealing with legislative auditing and accounting requirements—is also configured from this dialog box, on the Auditing tab, which Figure 5.13 shows.

**Figure 5.12: Examining special permissions on a folder.**

This screen helps you determine where inherited permissions are coming from (in the Inherited From column) as well as enables you to control the more detailed and granular permissions that can be associated with this particular object. Auditing—a key portion of any secure enterprise, especially those dealing with legislative auditing and accounting requirements—is also configured from this dialog box, on the Auditing tab, which Figure 5.13 shows.
Chapter 5

Figure 5.13: Auditing entries applied to a folder.

Unfortunately, even this dialog box doesn’t always display the full complexity of what has been applied to the folder. Figure 5.14 shows the dialog box used to add an auditing entry to the folder, and you can see that it offers several choices that aren’t as easy to see in the summary dialog box that Figure 5.13 shows.
Figure 5.14: Adding an auditing entry to a folder.

The point is to demonstrate that Windows’ user interface (UI) for managing permissions seems easy-to-use at first, but in fact can be less than intuitive and overly complicated when managing complicated file permissions. In fact, I blame this UI for many of the file server security issues I’ve run across—the interface just doesn’t make clear what is going on, making it easier for administrators to make mistakes when applying permissions and auditing entries to files and folders. It is also frustrating that Windows has no built-in searching functionality for permissions. In other words, strictly using Windows’ built-in tools, there is no way to discover which permissions a particular user has across all the files on a server—or across multiple servers.

So what is the solution? Obviously, a third-party tool with a better interface. Several exist, in fact, and many offer an interface that is comfortably similar to the Windows interface, making the solution somewhat more intuitive (see Figure 5.15). Such products offer an Explorer-like folder hierarchy and the security information in one view, making it infinitely easier to browse folders’ security. Inheritance is more clearly displayed, and a form of shorthand is used so that most special permissions can be seen right on the main screen (notice the last permission entry, which lists RXCsCf for Read, eXecute, Create subfolder, and Create file).
In addition, these tools can provide built-in search functionality, as Figure 5.16 shows, allowing you to quickly locate specific permissions across specified objects. This functionality is sorely needed in Windows, especially on file servers with tens of thousands of objects.
If you need more robust reporting capabilities, there are third-party tools that offer the functionality you need. Such tools collect security information into a centralized database, allowing faster reporting on that data. Figure 5.17, for example, shows a file permissions search for all objects that have explicit Full Control permissions applied.
Figure 5.17: Configuring a permissions search in ScriptLogic Enterprise Security Reporter.

Figure 5.18 shows the results of the search, which took only a few seconds to run because all the security information had already been collected into a central database.
Figure 5.18: Viewing an Enterprise Security Reporter security report.

Remember the worst nightmare I opened this chapter with? Enterprise Security Reporter is a tool that can help solve it, by showing you every file and folder in your organization that has permissions assigned directly to a user, rather than to a group, allowing you to get your NTFS and other permissions properly reassigned to groups.

Companies that make tools with these capabilities include ScriptLogic, which offers Security Explorer and Enterprise Security Reporter (ESR). In addition, Quest Software offers Quest Reporter, which includes share permission reporting from a centralized database, similar to the way in which ESR works. Companies such as BindView and NetIQ also offer solutions that ease enterprise-level file and folder permissions management. All of these solutions are intended to help make permissions management clearer and easier so that you can get the right permissions on the right files and folders. Of course, once you have the right permissions in place, the trick lies in keeping them right.
Maintaining Proper File Permissions

Understanding how to properly maintain file permissions requires an understanding of how they can get messed up to begin with. Once in place, properly applied file permissions don’t usually require a lot of maintenance. After all, they don’t get old and stale and require occasional refreshing; file permissions stick around for the life of the file. Where permissions can get messed up, however, is when the file gets moved or copied. When moving a file to another location on the same volume, Windows will retain the file’s permissions. When copying a file or moving the file to another volume, however, the permissions are reset to whatever permissions can be inherited from the new parent folder. This change is rarely what you want to do, however, so you will have to take special steps to ensure that file permissions remain in place.

One way to do so is to use the command-line Xcopy tool, which provides an /O command-line argument that copies file ownership and permissions. You can also specify the /X argument, which includes not only permissions and ownership but also auditing settings. Unfortunately, Xcopy can be a bit tedious if you’re copying a lot of files and folders, and it’s of no use when copying files and folders between servers.

Some administrators choose not to worry about it. Instead, they just copy the files and let the permissions get messed up, then either manually fix the permissions using Windows’ UI (or a third-party tool), or they use a command-line tool such as Cacls or Xcacls. Both of these tools allow you to replace or edit the file permissions on a file; Cacls—shown in Figure 5.19—is the simpler of the two.
Figure 5.19: Using the Cacls command-line tool.

And, simple as it is, Cacls is actually pretty complicated and somewhat limited in what it can do. Xcacls is more flexible and capable, but its command-line syntax is really complex. The bottom line is that, while both tools do a great job, they’re both pretty tough to use (I always have to look up some examples before using them) and they’re easy to make mistakes with. Enter a third-party tool to make the task a bit easier. As Figure 5.20 shows, a third-party tool lets you specify multiple files and folders to copy. By default, it copies permissions, auditing entries, and everything else, making the copy a simple operation whether you’re copying files to a new location on the same server or to an entirely different server.
One problem you can run into when copying files between servers, however, is the possibility that a user group that has permissions on the original server doesn’t exist on the target server. If you’re following one version of permissions-assignment best practices—that is, assigning permissions to servers’ local groups, and placing domain user groups into those local groups—this practice creates problems when copying files across servers. A third-party tool can deal with this problem by first checking to see whether the group exists on the target server and then, if you desire, creating it for you if it doesn’t exist (see Figure 5.21).

Figure 5.20: Using ScriptLogic Secure Copy to copy files.
Figure 5.21: Migrating local groups and users to the destination server.

Obviously, this difficulty is one reason I prefer to simply always assign permissions directly to domain user groups (which is really the best practice; assigning to local groups is often either a misunderstanding of the domain—best practice or a political necessity often based on departmental, rather than central, control of a file server). Once you’ve gotten file copying fixed—that is, you’re able to routinely copy files as needed without losing their permissions—you’ll have closed the loop on file server management, helping to ensure a more secure Windows enterprise.

Summary

File serving is one of the most common uses of Windows in any enterprise, making file servers worth special attention when it comes to security. This chapter covered how to make file permissions management easier and more efficient as well as how to keep file permissions intact during routine file maintenance and copying. I’ve discussed the role of storage management in helping to make file servers more secure, and we’ve explored key techniques that can help protect file servers from attacks. You’ll see many of these concepts again in the next chapter, applied more broadly (and covered in more detail) to securing Windows servers in general.
Chapter 6: Securing Servers and Services

The previous chapter showed you ways to make securing file servers a bit easier; this chapter will focus on security for all servers: Web servers, domain controllers, application servers, database servers, and more. Of course, all of these topics apply to file servers, too.

The crucial consideration to recognize about any server—regardless of OS, although I’ll focus on Windows—is that the OS contains bugs. Some of those bugs will create security vulnerabilities. Thus, the best practice for any server’s security is to try to shield those potential bugs and vulnerabilities, which is much of this chapter will focus on.

Lock it Down

Many administrators spend hours locking down their servers—removing unnecessary services and applications, blocking unnecessary TCP and UDP ports from external access, and so forth. As the Windows OS has matured, these tasks have become easier for administrators. For example, Windows 2000 (Win2K) Server includes the Internet Connection Firewall (ICF), which allows you to block access to unnecessary ports if it is difficult to disable or uninstall the software that is using these ports. As Figure 6.1 shows, Windows Server 2003 (WS2K3) Service Pack 1 (SP1) upgrades the ICF to the full Windows Firewall, providing more granular control and the ability to open ports based on tasks (such as file sharing) rather than forcing you to look up port numbers.

![Figure 6.1: Configuring the Windows Firewall.](image)

Figure 6.1: Configuring the Windows Firewall.
Unfortunately, all of Windows’ new capabilities are significantly hindered by the fact that much of Windows functionality requires multiple services to be running, necessitates multiple ports be opened, and are too poorly documented for administrators to accurately lock them down. Over the years, Microsoft has amassed a series of articles (such as the one at http://support.microsoft.com/default.aspx?scid=kb:en-us:287932 that covers Microsoft SQL Server) that attempt to detail the ports needed for various services; however, these efforts have never really come together into one comprehensive resource.

Security for Mere Mortals
Such was the case until WS2K3 SP1, which introduces the new Security Configuration Wizard. SCW is, in many respects, one of the best arguments for upgrading at least one server in your environment to WS2K3 SP1 (you must upgrade to SP1 in order to access SCW). As Figure 6.2 shows, SCW is based upon a set of XML-formatted configuration files that detail the various OS dependencies associated with specific server tasks or roles.

Figure 6.2: The SCW XML configuration file.

In Figure 6.2, for example, you can see a role named ExchangeBackEnd, which is an Exchange Server back-end server. This role has several dependencies: It depends on the presence of the role named Web, for example, as well as on a set of services and a series of TCP and UDP ports. The XML shipping with WS2K3 SP1 includes role definitions for most Microsoft server products as well as for common core roles such as domain controller, file server, and so forth. This XML file finally consolidates all of Microsoft’s previously scattered knowledge about which applications need which services and ports in order to function properly.
Because the XML file is well-documented and fairly easy to understand, you can easily update it to include dependency information for your own internal applications, and other software vendors can provide configuration information for their applications, as well.

All of this functionality allows SCW to lock down servers more accurately, providing what some experts refer to as “security for mere mortals.” If you’re running WS2K3-based servers in your environment, you can finally—with some confidence—lock them down to prevent yet-to-be-discovered bugs and vulnerabilities from being as big a problem.

How Big Is This Problem?

It’s easy enough to determine which ports are currently opened by your Windows servers. Simply run

```
netstat -a
```

from a command-line (or run

```
netstat -a -o
```

to see ports and their associated processes). As Figure 6.3 shows, a freshly installed standalone server opens quite a number of ports.

![Figure 6.3: Checking open ports on a Windows server.](image)

It is a good practice to document which ports should be open on each of your servers and check each server periodically to see which ports are actually open. For that task, netstat—which is designed to be run locally—isn’t the best tool. Instead, you might find an inexpensive (or even free) IP port scanner to be more effective, as it can scan entire subnets full of computers at once, reporting on open ports. Figure 6.4 shows the interface for a free scanner called Angry IP Scanner, available from [http://www.angryziber.com/ipscan/](http://www.angryziber.com/ipscan/).
Figure 6.4: The Angry IP Scanner interface.

Getting to know which ports are open on your servers, as opposed to which ones should be open, can help you understand the criticality of open ports: Every open port represents a potentially undiscovered security vulnerability. Although the tasks performed by a server certainly need to be accessible to clients, ports not related to those official tasks should be locked down to help prevent them from becoming a vulnerability.

Running SCW

SCW starts as Figure 6.5 shows, by asking you which computer will be the baseline for a new policy. SCW is designed to create multiple policies, with one policy assigned to each similar class of computers. For example, if you have five Web servers that perform substantially the same tasks, one of those could serve as your baseline for the Web server category of server.
Figure 6.5: Specifying a baseline server in SCW.

SCW will scan the baseline server and present a screen similar to the one that Figure 6.6 shows, listing all the roles currently installed on that server. A role is a business-level set of tasks, such as Web server, Telnet server, SQL Server, print server, and so forth. Notice that not all of the roles are selected, meaning the software supporting the role—such as print server—is installed, but the role isn’t enabled or in use (perhaps meaning no printers are actually shared).

Figure 6.6: Listing currently installed roles in SCW.

By making a different selection from the View drop-down list box, you can instead view currently selected roles (as Figure 6.7 shows) or all available roles (see Figure 6.8).
Viewing selected roles allows you to work with only the roles that are installed and in active use on the server; as any unselected roles aren’t in use, it is most likely safe to disable or uninstall them without negatively impacting your production environment. In contrast, by viewing all available roles, you can potentially use SCW to prepare a server for future roles that you know it will hold, but doesn’t currently. For example, if you know the server is going to become a member of a cluster in the future, you can prepare it for that role now, even though the necessary software hasn’t yet been installed.
SCW has a great deal of intelligence built in with regard to role dependencies. For example, the SQL Server 2000 Reporting Services role has a dependency on the Web server role. As Figure 6.9 illustrates, if I clear the Web server role check box—thus disabling it and blocking access to it—the SQL Server 2000 Reporting Services role check box automatically clears because it can’t function without the now-unavailable Web server role.

![Security Configuration Wizard](image1)

Figure 6.9: Automatic support for interdependent roles.

This support for role dependencies allows you operate SCW with more confidence. If you disable a role because you think you don’t need it, SCW will respond by disabling any dependent roles automatically. This action informs you that you did, in fact, need the role you disabled and prevents you from making a mistake.

Servers are also clients, so SCW next allows you to work with installed client features (see Figure 6.10). Again, you can choose to work with installed features, selected features, or all possible features, just as when working with server roles. In this step, you will decide whether the server will support Dynamic Host Configuration Protocol (DHCP), DNS, automatic updates, and so forth. Clear the check boxes of client software that isn’t used by the server.
SCW is also aware of administrative capabilities and other services, such as Alerter, audio services, and so forth. By selecting the appropriate options, you can ensure that the server is properly configured and that the software needed to administer or maintain the server will be enabled. Again, you can choose to work from a list of installed options (as Figure 6.11 shows), only those options currently in use, or all available options recognized by SCW.

SCW may also detect other services that don’t fit into the categories of server roles, client abilities, or administration and maintenance options. These last services will be displayed in a list (see Figure 6.12). For this list, you will often need to conduct extra research to determine what these services are doing, and decide whether you actually need them to be running on the server.
Next, SCW will ask how you want to handle any services that haven’t been explicitly selected in the prior lists. The default setting is not to change anything; however, I recommend configuring SCW’s policy to disable unknown services. Doing so will handle any services currently installed or any services installed in the future, ensuring that only those roles the server is supposed to be fulfilling will operate. Figure 6.13 shows the selection screen.

**Figure 6.13: Configuring additional services in SCW.**

**Figure 6.13: Choosing to disable unspecified services.**
Remember, however, that this policy is being constructed by looking at a baseline server. Say your baseline server is a Web server, and you’ve configured the policy appropriately. You have four other Web servers to which you plan to apply this completed policy. If one of those servers is also acting as a domain controller (for example), selecting Disable the service in this step of the SCW configuration process will disable the domain controller services when the policy is applied to that server. The reason is that the baseline server wasn’t a domain controller, so the domain controller services weren’t specified in the policy created by SCW. Thus, you need to be sure when applying a policy to any server that the server is identical in purpose to the server that was used as the policy’s baseline.

Near the end of the configuration process, SCW will display a list similar to the one that Figure 6.14 shows, detailing every service and which changes will be made by the policy. Notice the Current Startup Mode and Policy Startup Mode columns, which list the current state of each service as well as the state of each service once the policy is applied.

![Figure 6.14: Listing the changes to each service.](image)

Many services previously set to Manual startup will be set to Disabled once the policy is applied. This list provides you with the opportunity to perform a “sanity check” on the changes implemented by the policy. Until now, you’ve been dealing with roles and features, this list is the first time SCW has displayed its configuration in terms of direct changes to services. Similarly, the next screen (see Figure 6.15), provides a list of changes that will be made to TCP and UDP ports.
SCW will enable the Windows Firewall and will only create exceptions for ports that are selected in this list. You usually won’t need to make any changes, but it is useful to document this list as the approved list of ports for the server. You can also make adjustments to the port policy. For example, double-clicking a port allows you to modify its restrictions. As Figure 6.16 illustrates, you can allow a port to be completely open, restrict access to the port to a range of IP addresses, or require IP Security (IPSec) actions—such as encryption or authentication—to port access. For example, you might configure the policy to leave port 21 (FTP) open only to clients that can successfully establish an encrypted IPSec connection, thus protecting the FTP traffic—which transmits passwords and data in clear-text—from electronic eavesdropping.

**Figure 6.15: Examining port changes.**

**Figure 6.16: Configuring advanced port restrictions.**
This additional flexibility in SCW allows you to work with complex IPSec policies in a simpler, more intuitive fashion. Many companies like the idea of IPSec, but quickly become intimidated when they dive into Windows’ rather complex and unintuitive IPSec management interface; SCW helps make IPSec more approachable for those organizations. For example, you can restrict traffic based on a simple-to-configure drop-down box, requiring signing, encryption, or other options for incoming traffic on the specified port (see Figure 6.17).

![Figure 6.17: Easily configuring IPSec options within SCW.](image)

Next, SCW tries to get a feeling for the spare processor power your server has so that it can recommend additional security options. As Figure 6.18 shows, you can specify that the server doesn’t have any really old clients (really old being defined as Windows NT 4.0 before SP6a, Windows 95 without the Directory Services client, and so forth), and you can indicate that the server has some spare processor power. If you select both options, SCW will configure the policy to digitally sign all file and print traffic, helping to prevent traffic spoofing. This setting consumes about 20 percent extra processor power, so only select this option if your server is running at 70 percent capacity or less.
Configuration Wizard

**Figure 6.18: Configuring traffic signing in SCW.**

SCW isn’t just asking whether you want to digitally sign traffic; by asking you whether the computer has spare processor power and whether it needs to support older clients, SCW is alerting you to the requirements of digital signing, and helping prevent you from making a decision that isn’t suitable for your environment. Similarly, as Figure 6.19 shows, SCW will ask which types of accounts are used by the server to authenticate with remote computers. Your answer will help configure the Local Security Authority to the highest possible level of authentication.

**Figure 6.19: Configuring outbound authentication.**
SCW isn’t simply asking to which level outbound authentication should be set (which is what you would need to decide if you configured this setting in, for example, a Group Policy Object—GPO). Instead, SCW is helping you make a decision based on information you’re more likely to know and understand, such as which user accounts are utilized.

Not sure when a server would make an outbound connection? When the server is talking to domain controllers, for example, perhaps sending data to a network-attached printer, copying files from other servers, and so forth.

Another example of this more intuitive approach is when SCW gathers additional information to fine-tune local security (see Figure 6.20). In the previous screen, I indicated that remote local accounts are used, so SCW needs to know if those remote computers are at least running NT 4.0 SP6a and whether they synchronize their clocks (remember, Windows’ Kerberos authentication requires synchronized clocks). This information helps SCW select an appropriate authentication scheme.

You might have noticed small blue text links at the bottom of many of SCW’s screens. These links open a Help file to help you learn more about the settings being configured by that screen of SCW. This link is a great way to learn more about the actions that SCW is performing based on each screen’s selections.

SCW isn’t limited to securing ports and services, however. As Figure 6.21 shows, the prior few screens affect registry-based configuration settings.
Figure 6.21: Checking registry settings in SCW.

Just as SCW allows you to review its low-level decisions in services and ports, it now allows you to review proposed changes to registry settings before continuing. This process is all part of Microsoft philosophy reflected in SCW of asking you higher-level questions, but still allowing you to review the final low-level configuration decisions.

Next, SCW will configure auditing (see Figure 6.22). The default setting is to audit successful activities, providing what Microsoft calls a *high signal-to-noise* ratio, meaning you’ll get a lot of good auditing data without a lot of garbage. You can (as shown) change this setting based on your organization’s requirements.

Figure 6.22: Configuring auditing.
As always, SCW takes your decision and proposes a number of configuration changes, which it then shares with you as Figure 6.23 shows.

**Figure 6.23: Reviewing the changes to the audit policy.**

Finally, as Figure 6.24 shows, you’ll be asked to save the policy in an XML file. You should provide a detailed description indicating for which type of server this policy is intended; a detailed description will help prevent confusion and error when finally applying the policy.

**Figure 6.24: Saving the new policy.**
SCW is not necessarily configuring the server on which it is running. Remember, you specified a baseline computer at the start of SCW, and the policy is built for that server. The final screen of SCW (see Figure 6.25), you can choose to simply have the policy saved to a file or you can go ahead and apply it now. You can always re-run SCW later, load in a previously saved policy, and apply it to another computer.

⚠️ Applying a policy typically requires a server restart because changes to services’ startup modes won’t take effect until the server is restarted and those services all stopped.

![Figure 6.25: Choosing when to apply the policy.](image)

**The Results**

As Figure 6.26 shows, SCW does have an effect. I chose to apply the policy immediately, and you can see by the `netstat -a` command output that fewer ports are now open. For easy comparison, this figure shows the ports open before SCW was run (on top) as well as after (on the bottom).
SCW provides an effective way to configure multiple servers in a more locked-down fashion, while helping you to avoid typical mistakes that arise from Windows’ inherent complexity. However, although SCW offers a great start to configuring high-level security, it doesn’t attempt to deal with a number of crucial details.

**Beyond the SCW**

SCW is primarily intended as a one-time operation, meaning you’ll apply a policy to a server and then let it sit. SCW doesn’t attempt to handle what you might call day-to-day security tasks, such as managing service identities and passwords; managing registry, file, and folder security; and so forth. Unfortunately, although these tasks are more immediately comprehensible than locking down ports and services, they’re made more difficult by the sheer size of the task. I’ll cover them in the next two sections.

![Figure 6.26: Comparing ports from before and after applying the SCW-created policy.](image)
Service Security

This guide has previously declared the critical need to maintain service security: ensure services aren’t running under accounts that have excess authority (running every service on a server under a domain administrator account is an all-too-common scenario) and that services’ passwords are changed on a regular basis. Both of these tasks are difficult to accomplish across a number of servers. For example, changing the password used by a service is easy on one server; it’s time-consuming and error-prone when you’re faced with a dozen or more. Scripts can help automate the process; for example, the script that Listing 6.1 shows is from the Microsoft TechNet Script Center and changes the password of a single service on single computer.

```vbscript
strComputer = "." Set objWMIService = GetObject("winmgmts:" & "{impersonationLevel=impersonate}!\\" & strComputer & "\root\cimv2") Set colServiceList = objWMIService.ExecQuery _   ("Select * from Win32_Service Where StartName = '.\netsvc'") For Each objService in colServiceList    errReturn = objService.Change( , , , , , , , "password") Next
```

Listing 6.1: Changing the password of a single service on single computer.

It’s a simply process to convert this script into one that runs against multiple computers, and many administrators use techniques such as this to automate their environments. What a script can’t readily do, however, is ensure that it has caught every single service that needs its password updated. If you miss one, it won’t start the next time it needs to, and you’ll wind up with a Help desk call and upset users. Similarly, a script can easily be used to list all services running with a particular user account, provided you point the script to every server. Miss one, and you’ll have incomplete information. In short, scripts are great for well-maintained, well-documented environments—but you may not have one of those.

You’ll find other services-related scripts at

Commercial tools are often a better way to work with services in larger environments. For example, Lieberman Software’s Service Account Manager (see Figure 6.27), provides a single view of all services on all systems, allowing you to make changes and check services more easily.
Another product, ScriptLogic Service Explorer, provides similar capabilities. As Figure 6.28 shows, you can easily view all the services on any remote system, make changes to services’ configurations, and so forth. Service Explorer also handles the management of Scheduled Tasks, which also often run under a user’s security credentials.
Effective in large environments is the product’s built-in search capabilities. For example, as Figure 6.29 shows, you can right-click any service and add it to current search criteria (the “add to search” option is at the bottom of the context menu). This feature makes it easy to quickly find all servers running that particular service, making it easier to then change the password or account used by the service, or even to change its startup type or other configuration parameters.

Figure 6.28: Listing services with Service Explorer.
Searches can be performed against any scope, such as a manually constructed list of servers, your entire network neighborhood, a workgroup, or even—probably the most useful scope—an entire domain. Search capabilities are broad: you can specify search criteria (shown in the window’s upper-left corner) that includes service names, account names, and so forth. As Figure 6.30 shows, the search process will even include detailed error messages indicating servers that couldn’t be reached, ones for which you haven’t specified valid logon credentials, and so forth.
Products such as Service Explorer and Service Account Manager make it easier to manage services and their security across a large number of servers. You can easily change service passwords on a regular basis, reconfigure services to run under the right user accounts, locate services using a particular user account, and so forth. As many of these tasks—especially updating services’ passwords—go ignored in most environments, these tools are effectively enabling a much higher level of server security than you likely already have.

Access Controls

The previous chapter spent a lot of time on techniques to better manage file and share permissions. Of course, access control isn’t limited to file servers; all servers have files and folders that need to be secured. They have registries, too, which are an often-overlooked area of security that can lead to significant security compromises. Keep in mind that most Windows software stores all its configuration settings in the registry; a compromised registry can lead to seriously compromised applications and servers. With that in mind, let’s explore access control as a general topic, focusing on tools and techniques that can address both files and folders as well as the registry.
Finding Permissions

Windows’ built-in security tools—including its graphical access control list (ACL) editor and command-line tools such as Cacls.exe—provide effective basic capabilities for working with access controls. However, these tools lack any kind of search capability. For example, if you want to find all files on which the special Everyone account has permissions, doing so is difficult without finding a third-party tool. Fortunately, third-party tools are readily available to handle this task. For example, Figure 6.31 shows such a tool being used to construct a search. As shown, the permissions from and existing folder are displayed, and the Everyone group—which has permissions on that folder—is selected. Clicking Begin Search starts the search for other files and folders with that group assigned.

![Figure 6.31: Searching for the built-in Everyone group.](image)

The results of the search, which Figure 6.32 shows, quickly identify files and folders where that selected group has permissions, and shows you which permissions are assigned. For example, D:(RX) F:(RX) indicates that the group has Read and Execute permissions on a folder and on files within that folder.
If you find permissions you don’t like, third-party tools make it easier to get rid of them. As Figure 6.33 shows, for example, you can easily get rid of the Everyone group throughout a directory tree. I’ve selected the special Everyone group and indicated that I want to Revoke All Permissions for the user. This action will—and here’s where Windows’ security terminology can become confusing—remove any “allow” and “deny” permissions, effectively erasing the Everyone group from the ACLs.
Figure 6.33: Revoking permissions for the Everyone user.

Because server permissions are a bit simpler on non-file servers (typically, a single set of permissions suffices for entire volumes), you can use third-party tools to fix the permissions on a server fairly quickly. Plus, certain tools manage registry permissions as well (see Figure 6.34, which shows an almost-identical screen to Figure 6.33 in which you can correct registry permissions for an entire registry hive or key).
Figure 6.34: Correcting registry permissions.

Many enterprise environments focus on file and folder security on file servers but don’t worry about it much on their other servers. After all, users aren’t usually given access to the file system on non-file servers, so why bother? In addition, managing file, folder, and registry permissions can be time-consuming. However, with the right tool, permissions can be set up much more effectively with much less effort; taking the time to do so also provides an additional layer of defense in case the wrong person does gain access to the file system or the registry.

Permissions Reporting

Being able to report on permissions is a useful capability, especially in today’s audit-heavy world of legislative compliance issues. Third-party tools offer the capability to export permissions to an Open Database Connectivity (ODBC) database, such as Microsoft SQL Server. Once in the database, you can create customized reports using standard reporting tools such as Crystal Reports or SQL Server Reporting Services. Figure 6.35 shows the export process.

ScriptLogic makes another product, Enterprise Security Reporter, which is better-suited for large-scale permissions reporting. Other companies such as NetIQ and Ecora also offer security-reporting solutions.
Figure 6.35: Exporting permissions to a database.

Backing Up Permissions
Any backups you make should also back up your permission sets. Windows Backup and most third-party backup utilities are all capable of backing up permissions along with files, and usually handle the registry and its permissions, as well.

Sometimes, however, you might want to restore only the permissions on a file, folder, or registry key. Most backup software can only do so if you’re also restoring the file, but you may have occasions where you want to restore the permissions—because, perhaps another administrator configured them incorrectly—but not the file, folder, or registry key (perhaps it has changed since the last backup was made). Conveniently, effective third-party tools provide this capability by allowing you to back up and restore only permissions (see Figure 6.36).
Summary

Server security boils down to a few basic areas:

- Access controls
- Service management
- Lockdown (of services, applications, and ports)
- Authentication security

Unfortunately, these areas are either large and difficult to manage or complex and difficult to accurately configure (or both). By using tools the tools that this chapter introduces, however, you’ll be able to more easily deal with mass-security reconfigurations, service management, and server lockdown.

Although locking down unnecessary services is a great step to take toward server security, unauthorized or unnecessary software—not just services—is one of the biggest problems facing today’s enterprises. The next chapter will talk about software management, software maintenance, and software filtering, techniques guaranteed to help make your Windows enterprise more secure.
Chapter 7: Security Through Software Maintenance and Filtering

Before computers became connected through the Internet, software wasn’t such a scary thing. Sure, viruses existed, but they were definitely limited in their effect by the difficulty of multiplying between disconnected computers. Software in general, in fact, has only recently started to become routinely complex. Ten years ago, the only folks worried about “patches” and software maintenance were typically server operators in large environments, perhaps working on systems such as an IBM AS/400 midrange. Today, however, complex OSs and all their associated maintenance needs live on every desktop, portable computer, and PDA. The high level of connectivity between these systems makes viruses, spyware, and adware—collectively referred to as malware—easy to catch and easier to propagate.

As a technology professional, you’re probably more than a little tired of hearing about patch management; I know I am. In this chapter, however, I want to put patch management in the overall context of software management, and discuss software management—the process of handling everything about all the software in your environment—from a much broader perspective. In keeping with the theme of “overlooked” security issues, I want to focus on software management issues that are often overlooked, or that don’t get the full attention they deserve.

What is Software Maintenance?

The idea behind software maintenance is that software grows old. Bugs are discovered and fixed, features are improved, and so forth; maintenance is the process of keeping the software up-to-date. Fairly recently, we’ve started to categorize maintenance into two broad, but distinct, areas: critical maintenance and everything else. Critical maintenance typically involves maintenance—such as patches—that help make the software more secure or significantly more reliable. With all the people in the world releasing malware designed to exploit discovered vulnerabilities, critical maintenance is becoming…well, more critical than ever.

This guide is about securing Windows in the enterprise, so it will understandably be focusing on that platform. However, just about everything here applies to every OS. No software is inherently more secure than any other software, whether it’s commercial, free, open source, shareware, homegrown, or any other variety. All software has bugs, some of those bugs are security-related, and they will all require maintenance at some point.

Patches, Patches, Patches

It used to be that you could just wait for Microsoft to release an OS or application service pack, wait a couple of months beyond that for the bugs in the service pack to be fixed, apply the service pack, and your systems would be updated. Unfortunately, those days are long gone. Security vulnerabilities require rapid action because they’re often exploited nearly as quickly as a patch can be developed.
Windows definitely has a less-than-stellar reputation when it comes to security. That’s perhaps undeserved, because most software is less-than-stellar when it comes to security. However, Windows’ bad reputation has had one great benefit for server administrators—convenience. In an attempt to help bolster its OS’s reputation and acceptance, Microsoft has done more to make patch management easy and understandable than any other OS vendor.

Patch management is a common point of discussion and not overwhelmingly exciting. Thus, this section will simply help you spot some of the problems with patch management—ones you may not even have articulated to yourself—and suggest some ways to solve them.

Must I patch?
Yes. Resoundingly, yes. The following list highlights several good reasons, in fact:

- **It’s your job**—The first and foremost priority in any organization should be to keep existing resources well-maintained, and patches are a big part of that maintenance.

- **Compliance**—If you’re in an organization that’s subject to a legislated security or accountability standard (and most companies are, these days), the auditors are going to be paying close attention to patch management. In the IT community, we can’t test the “end state” of security. In other words, if you say “such and such a file can only be accessed by authorized individuals,” you don’t really know that the statement is true unless everyone tries to access the file. Thus, you rely on configuration settings to provide security. The problem is, of course, that the configuration settings are only as good as the OS—if you have a bug in the OS, the configuration might not be properly enforced. Thus, applying patches (from a security auditing point of view, at least), is the only way to make sure that your configuration is being properly enforced.

- **Exploits**—The people who write malware are bright. The last several exploits of Windows vulnerabilities, in fact, have been reverse-engineered *from the patches*. Microsoft quietly announces a vulnerability, releases a patch, then spends a month marketing the patch so that administrators will install the patch. During that month, the virus-writers download the patch. They compare the patched version of files to the unpatched version, looking for differences. They then disassemble the code and look at the areas that are different to see what changed. Based on that research, they write an exploit that affects unpatched systems because they know that most systems will remain unpatched until the next service pack comes out. Slammer and Blaster are examples of effective viruses that were released *months* after a patch was available.

Thus, there are plenty of reasons to install patches and to do so in a timely manner. Two major problems often prevent that from happening: The time crunch and organizational policy.
Should I Test Patches First?

The answer to this question had it been asked a year ago would have been “of course.” The answer is no longer so clear. Many organizations have a must-test-first policy, but they’re so understaffed and overworked that the testing is put off. The result? Great intentions and unpatched systems. The new theory is that you shouldn’t test critical patches first. No matter how badly Microsoft might have screwed up the patch (and in this regard, the company has a pretty good reputation—very few patches are re-released due to bugs) the effect won’t be as bad as getting hit with the exploit.

Consider this new theory—companies implement this test-patches-first policy because they are afraid of the patch breaking something. In the meantime, their systems sit unpatched waiting for an exploit that will quite definitely break things. And the “we’re protected by firewalls” argument doesn’t hold up because every organization hit with Slammer, Blaster, and the other great exploits of the past all had firewalls.

If you have a choice between systems that only contain tested patches but are running weeks behind the patch leading edge or having systems that contain the latest—yet untested—patches, the best option is the latter. At least then if there is a problem with a patch, you can call Microsoft product support. If you get hit with an exploit, you have no one to blame but yourself.

Windows Software Update Services

To help combat the time crunch problem, Microsoft released Software Update Services (SUS); Windows SUS (WSUS) is essentially a much-improved version of the SUS software. The basic theory behind WSUS is that all of your machines have the Automatic Updates client already installed (it is in the latest service packs for Win2K and later); most systems are configured to look at the Windows Update Web site to obtain patches. Realizing that organizations want a bit more control over incoming patches, Microsoft leveraged Automatic Updates to make WSUS a sort of “corporate Windows Update.” WSUS is designed to be deployed in a hierarchy of servers, as Figure 7.1 shows.
As shown, your top-level WSUS server (or servers) downloads updates from the Microsoft Windows Update Web servers. At that point, updates are held pending your approval. You can create groups for your computers, and approve updates on a group-by-group basis. This feature allows you to, for example, approve updates for a small group of test machines, then later approve updates for all of your computers. Clients’ Automatic Updates software is configured to look to your WSUS servers rather than Windows Update, for their updates; this configuration can be centrally managed through Group Policy. You can deploy additional WSUS servers that receive their updates from your top-level WSUS servers, creating an internal hierarchy.

Some bandwidth savings is achieved by this deployment. For example, if you deploy a WSUS server at each geographic location in your company, local clients can receive their updates without utilizing WAN bandwidth. WSUS itself uses Windows’ Background Intelligent Transfer Service (BITS) to download updates, which helps to minimize bandwidth utilization by “trickling” updates down to the server. You can use Group Policy (in, for example, site-linked Group Policy Objects—GPOs) to direct clients to the proper WSUS server in your organization. Update approval can be centralized at the top-level WSUS server, minimizing management overhead.
WSUS also provides impact analysis functionality, allowing you to determine which computers, if any, would need a selected update. WSUS accomplishes this by polling the Automatic Updates software when it checks in for updates, and compiles an impact report for you. WSUS also tracks patch deployment—you can see which computers have received a given patch. For critical patches, you can allow users to defer installation (useful if the patch requires a restart), but still set a “drop dead date,” after which Automatic Updates won’t allow installation to be deferred any longer.

For a free tool, WSUS is an effective tool. In fact, it’s probably one of the best administrative tools Microsoft has released, free or otherwise. It integrates well with Microsoft Systems Management Server (SMS), but it does have some functional limitations, which is why many organizations still choose to deploy commercial patch management solutions.

**Microsoft Baseline Security Analyzer**

Another free tool from Microsoft, the Microsoft Baseline Security Analyzer (MBSA) is designed to scan the local computer, or one or more remote computers, and report on a number of security areas. One major area is patches; other areas include security-related configurations, and so forth.

> As of this writing, MBSA 1.2.1 is the latest version. However, MBSA 2.0 just wrapped up its beta and will be released in the summer of 2005.

MBSA is based on code written by Shavlik, which offers a commercial version of the product called HFNetChk. MBSA uses an XML-formatted database, maintained by Microsoft, which contains information about security patches, settings, and so forth. The most current version of this database is downloaded by MBSA automatically whenever you run the tool, and provides MBSA’s scanning engine with the information necessary to evaluate your systems.

> Prior to version 1.2, Microsoft’s various scanning tools—Windows Update, Automatic Updates, and MBSA—used different scanning engines and could return slightly different results. MBSA 1.2, WSUS, and the current Automatic Updates client use the same scanning engine to provide more consistent results. This new scanning engine also supports Microsoft server products (such as SQL Server and Exchange Server) as well as business applications such as Microsoft Office.
As Figure 7.2 shows, MBSA highlights missing security patches for a number of products, helping you quickly determine whether you’re up to date.

*Figure 7.2: Viewing MBSA results for security patches.*
However, MBSA isn’t limited to patch management (see Figure 7.3).

In this example, MBSA has spotted potential problems with the system security as well as security configurations in IIS that don’t meet best practices. MBSA’s report is fairly comprehensive, but the tool does have some limitations. First, MBSA is a report-only tool; it does nothing to fix security problems. It’s also not ideal for scanning large numbers of computers because it does so entirely over the network and can take a long time. The reports produced for multiple computers aren’t ideal for management because they take effort to determine which machines need which fixes. Clearly, MBSA is an effective tool for a periodic checkup of key machines but falls short in the area of long-term, proactive management.
Commercial Patch Management

Several companies offer commercial software solutions specifically designed for patch management. One example is ScriptLogic’s Patch Authority Plus, and another is ConfigureSoft’s Enterprise Configuration Manager (ECM—which is really a configuration management product; a plug-in solution called Security Updates Manager—SUM—adds patch-management capabilities to ECM). Of course, the aforementioned HFNetChk is also a viable offering, and is available in various editions for different types of organizations.

ScriptLogic’s Service Explorer product is bundled with Patch Authority Plus—hence the “Plus.”

Commercial patch management tools usually take an agent-based or an agentless approach. Both approaches have their fans and critics; the option that works best for your organization depends on your needs and preferences. In an agent-based scenario, you have a small software agent that lives on each managed machine. This setup is similar to WSUS, where the built-in Automatic Updates client serves as the agent. Agent-based solutions tend to consume less network bandwidth because the agent can do a lot of work in data-collection and can compress and filter data before sending it up to the server. However, you have to deploy that agent—although some solutions can help do so automatically. The agent becomes an additional piece of software to manage, but some solutions help manage it for you.

In an agentless architecture, no software resides on managed machines. This setup is similar to MBSA. A downside is that more network bandwidth is typically consumed because the entire scan is done from a remote computer over the network; however, there is no additional software to deploy and manage. Agentless solutions typically get you up and running more quickly because there is no deployment required.

Interestingly, many commercial solutions are built on Microsoft’s XML database, used by MBSA. As a result, though, you’ll be comparing patch management solutions on their features and ease of use, and not so much on their ability to find patches, because many of them are using the same underlying data. Some solutions even leverage existing technology more—Patch Authority Plus, for example, utilizes Shavlik’s HFNetChkPro engine, providing a friendlier and more feature-rich administrative interface on top of that core engine. A benefit of utilizing HFNetChkPro is that it’s the same engine (essentially) powering MBSA, so you’re getting results consistent with Microsoft’s own tools.
Commercial solutions can provide a lot of value around the core patch- and security-management concept:

- **Organization**—Being able to group machines and manage them as a group is a big benefit in large organizations. Rather than deploying patches to 37,000 machines, you deploy patches to a handful of groups.

- **Business rules**—Adding business rules can make management easier, too. For example, you can define a set of patches as your “baseline,” then scan machines to determine compliance with that baseline. You’re therefore no longer looking through reports for individual patches, you’re looking for a “yes/no” indication of compliance.

- **Knowledge**—Rather than just allowing you to select patches, third-party solutions provide information about the severity of whatever the patch fixes, information about the files and registry settings that are updated, and so forth.

- **Centralization**—Some third-party products centralize deployment information into a single SQL Server database, providing powerful reporting tools (including email notifications), executive summaries, and more. These can be viewed through a Web portal or through any standard reporting package.

- **Customization**—Certain solutions don’t limit you to Microsoft’s patches; you can define your own patches using a documented XML format. Thus, you can use the commercial solution to patch third-party and homegrown applications as well as Microsoft’s software.

Figure 7.4 shows a third-party product’s scan summary screen, which provides an overview, showing top ten missing patches, systems that couldn’t be scanned (often due to improper credentials; some products allow you to provide default credentials for each machine group and provide per-machine credentials when necessary), and so forth.
Thus far, this example hasn’t illustrated that third-party tools can provide information much beyond what MBSA might provide for free. However, the difference is that some third-party products allow you to fix the problem, making management easier.

For example, you can build a patch group, which contains one or more patches. Unfortunately, selecting these patches can be time-consuming, because Microsoft hasn’t provided any friendly way of identifying patches beyond its MS00-086 style identification numbers or Knowledge Base article numbers. At best, you could identify patches by something like “RDISK Registry Enumeration File Vulnerability,” but the fact is that there are many patches and no easy way to remember which is which. Once you’ve built a patch group (or an individual patch, for that matter), a third-party tool can show you more detailed information about it—its description, such as “Web Server Folder Traversal Vulnerability;” its severity; whether it’s still available (many patches supersede earlier ones); and for which product the patch has been released. Figure 7.5 shows an example patch details screen.
When it comes to patch distribution, a third-party tool can help spread the workload by working with distribution servers that you define. This setup essentially allows you to enlist additional computers in the patch file-copying process, spreading the workload for a large organization. Ideally, each geographic location would have a distribution server because this method minimizes WAN utilization. Practically, one server can support about 2500 machines, so larger locations would benefit from more servers. Similar to AD site configuration, each distribution server can be assigned IP address ranges for clients, helping to ensure clients are services by a local distribution server. Patches can be deployed on a schedule (such as over the weekend) or immediately; you can define restart options (when needed), such as forcing machines to restart after installation.

Whichever solution you choose, you can see that commercial solutions offer capabilities above and beyond MBSA’s mere reporting—and even beyond the useful features built-in to WSUS (for example, WSUS doesn’t provide patch grouping and baseline compliance reports).
Security Software Maintenance

Another major area of software maintenance is the maintenance of security software, such as antivirus software. Although not strictly patches, the virus (or spyware or adware or whatever) definitions that allow this software to detect threats must be updated frequently. The underlying engine must be occasionally updated, too, to help detect entirely new breeds of threat.

Typically, even desktop antivirus software comes with some form of built-in automatic updating capability, requiring only a network connection to work. However, investing in solutions specifically designed for enterprise use can give you more control, such as through a WSUS-like internal-updating hierarchy, reporting, and so forth. For example, Computer Associates’ eTrust AntiVirus for enterprises includes centralized administration, reporting, and update distribution. It’s a cross-platform product, making it suitable for large organizations with multiple OSs that want to standardize.

The Windows Security Center

Windows XP Service Pack 2 (SP2) introduces a new Security Center, which provides a centralized view of key security measures. Although not strictly useful as an administrative tool, the Security Center does help make end users more aware of their overall security posture. Of particular interest is the antivirus indication. As Figure 7.6 shows, the Security Center will alert you when antivirus software isn’t installed; a “yellow” indication will appear whenever the software’s definitions appear to be out of date.
The Security Center doesn’t currently recognize anti-spyware or anti-adware solutions; it’s likely a future version of it will deal with these products individually, as Microsoft is entering the anti-spyware market.

Security Center requires that solutions properly register themselves (through a special Windows Management Instrumentation—WMI—interface) in order to be detected; thus, it’s possible to have a perfectly good solution installed but still get a “red light” in the Security Center (the version of Norton AntiVirus available at the time Security Center was introduced wasn’t properly detected, for example). Undetected applications are a difficulty because they force Security Center to give a “not protected” indication to end users, when they might be protected just fine. The bottom line is that you should go with a solution that can be detected because Security Center will help automatically alert your users to out-of-date definitions. If you’ve got an automatic updates scheme in place for your antivirus solution, you will want users to call the Help desk if Security Center starts complaining about out-of-date definitions, because it means a problem has occurred with your updates.
Boundaries and the Multi-Layered Defense

Never forget the key to successful security—protection at every boundary, creating a multi-layered defense. There is no question that you’re in a war against malware authors, and any smart general knows to have a backup plan. Or two. Or three. Deploy firewalls at your network boundaries. Deploy firewalls internally between network segments to help control the internal propagation of malware. Deploy firewalls to every layer of your network that can have a firewall, from clients to your ISP connection. Deploy virus scanners everywhere, too—at the boundary, on client computers, in your email infrastructure, in your proxy servers, and in your firewalls, too, if you can. The more likely you are to catch everything. Yes, your software management burden will increase, but your safety and security will increase by a much greater factor. For example, consider the network that Figure 7.7 shows.

![Diagram of a typical corporate network](image)

**Figure 7.7: A typical corporate network.**

This network has a number of typical elements: multiple subnets containing client computers, a subnet for servers, and a perimeter network (or demilitarized zone—DMZ) containing a Web server. A router connects everything; typically, firewalls protect the internal subnets as a group, preventing outside, unauthorized access. The firewalls are likely configured to allow only a small amount of traffic in (perhaps only HTTP traffic to the Web server), and configured to allow a broader range of traffic out. Figure 7.8 shows a more secure configuration for this network, employing a multi-layered defense.
Figure 7.8: Adding a multi-layered defense.

In this network, every device capable of acting as or running a firewall is doing so. The router, in particular, can be used to filter traffic between internal subnets. Although all traffic might be allowed to and from the server subnet, traffic between client subnets might be heavily restricted (because in many corporations client computers don’t share files or printers, there’s often no reason for them to access one another). Should malware infect one of the client computers, its scope will be severely limited because it won’t have immediate access to much of the network.

Microsoft has announced that Longhorn, the next major version of Windows, will feature a two-way firewall, providing administrators with the ability to restrict the traffic that can leave a computer as well as the traffic that can enter the computer. Such a firewall would help to further restrict the effects of malware on a network.

Of course, a similar technique should be used for any security measures, such as antivirus software—run it on every level possible.

What is Software Filtering?

Software filtering is a frequently overlooked aspect of software management. Most corporate environments simply allow any software installed on their machines to run; they rely on security permissions to prevent users from installing unwanted software, but unfortunately software comes in too many varieties. For example, users in a locked-down environment might not be able to install the latest game on their computers, but they can generally install Browser Helper Objects (BHOs), which are a common vehicle for spyware and adware. Software filtering can help better control the software allowed to execute in an environment, making it more secure.
The theory behind effective software filtering is to identify the software that *should* be permitted in your environment, then deny everything else the ability to execute. I don’t want to underestimate the enormity of that task in some environments, which may have thousands of permitted applications. But think about it—if you can positively identify every bit of authorized software, then successfully prevent anything else from executing, you’ll have eliminated most attack vectors used by malware and any other unauthorized application. Unlike an antivirus scanner, which seeks to identify unauthorized software (and can become out-of-date and fail to identify new ones), you’ll simply be disabling any software you haven’t heard of. That’s a much easier task and requires little long-term maintenance, making for a much more secure environment.

**Software Restriction Policies**

Software Restriction Policies (SRPs) were introduced in Windows XP and exist in WS2K3. Their purpose is to identify software by various means, then apply an “allowed” or “disallowed” security level to that software. By default, SRPs are configured with a default security level of Unrestricted (see Figure 7.9). This default allows all software to execute, unless you’ve identified a piece of software and applied the “disallowed” security level to it.

![Local Security Settings](image)

*Figure 7.9: Default SRPs.*

However, as I mentioned earlier, trying to identify all the *bad* software in the world would be practically impossible. Better to switch the system to use Disallowed as its default, then simply identify the software you *want* running on your computers.

⚠️ Don’t switch the default security level to Disallowed without taking some steps first or everything will quit working!
SRP does have a few options you should explore first, though. The first one is the Enforcement policy. By default, this policy causes SRP to only monitor executables, not DLLs. Because DLLs are called by executables, monitoring at the executable level is generally sufficient. Also, by only monitoring executables, your task of identifying permitted software will be much easier. You could, for example, simply identify Excel.exe as allowed, without worrying about the umpteen-billion DLLs used by Excel. You can also choose to have SRP apply to all users, or to exempt administrators.

The example screenshots are from a Windows XP computer’s local system policy. However, SRP can also be configured at the domain in a GPO, providing centralized software filtering for the entire enterprise.

Speaking of “executables,” a second SRP option defines what SRP considers to be an executable. Figure 7.10 shows a portion of the default list, which includes Control Panel extensions, various scripts, and so forth.

Finally, the last SRP option—Trusted Publishers—leads to a critical set of decisions. By default, SRP allows end users to determine which publishers are trusted. Change this setting to local admins or enterprise (domain) administrators.
Software can be digitally signed. Examining the properties of Excel.exe, for example, you can see that it has been digitally signed by Microsoft.com (see Figure 7.11).

![Figure 7.11: Digital signature on Excel.exe.](image)

So who published the certificate that was used to sign Excel.exe? By clicking Details, you can examine the certificate’s certification path to see where it came from. As Figure 7.12 shows, the publisher of this certificate is the Microsoft Root Authority.
So is Microsoft Root Authority a “trusted publisher?” To make this determination, open the Internet Options Control Panel applet, and click Publishers on the Content tab. You’ll see a tab named Trusted Publishers as well as one named Trusted Root Certification Authorities. Everything listed on either tab is considered a trusted publisher; Figure 7.13 shows that Microsoft Root Authority is indeed listed.
Figure 7.13: Viewing trusted publishers.

So what does that mean? It means any software that has been signed using a certificate issued by one of these trusted publishers is a trusted piece of software. This concept is central to the way SRP works. Microsoft pre-installs more than one hundred root certificates with Windows.

Take a second to understand the value of a trusted publisher. Let’s say a virus shows up in your environment and it has been digitally signed using a certificate issued by, say, Joe’s Certification Authority. If that virus does damage, you’ll be able to look at its signature to see who authored it, turn that information over to the authorities, and nab the offender. Of course, if the certificate issuer (Joe) didn’t do a good job of validating the person’s identity prior to issuing a certificate, then you won’t be able to catch the bad guy. It’d be as if your state issued drivers’ licenses without checking any other form of ID—as a form of ID, the license would be useless. Thus, this list of trusted publishers is, in essence, the list of people you trust to vouch for other people’s identities.

Let’s take a look at that list again. I trust VeriSign, for example, because I’ve taken the time to research its certificate-issuing policies, and I’m confident that the company makes it difficult for someone to obtain a false certificate. For any “trusted” root authority you haven’t investigated, you should think about removing them from the list to better protect your organization.
Removing a trusted publisher or certification authority means that publisher’s certificates will be considered invalid by your computer. This might prevent users from reaching “secure” Web sites who have certificates issued by the publishers you removed. You need to consider this decision—if you don’t trust the publisher, than the Web site wasn’t all that secure to begin with.

Okay, so how does all this apply to SRP? Through its additional rules. These are the rules that identify software that should be exempted from the default security level. SRP comes with four path rules, which exempt all software in the paths used by Windows (such as C:\Windows and C:\Windows\System32). These four rules (see Figure 7.14) define a level of “unrestricted.” Why bother, when that’s the default? It’s basically a safety mechanism—if you change the default level to Disallowed, these four path rules will keep Windows running. However, path rules are a bad idea because any software capable of copying itself into that path will run. Yes, path rules are easy—but not particularly secure.

![Local Security Settings](image)

Figure 7.14: The four default path rules in SRP.

My preferred types of rules are hash and certificate. A hash rule, which Figure 7.15 shows, identifies a particular executable by calculating a unique hash based on the executable’s file. If the executable changes in any way, the hash won’t match. This hash identifies Notepad.exe, and marks it as Unrestricted, meaning it’ll be allowed to run.
Many core Windows files are digitally signed by Microsoft; some—such as Notepad—inexplicably aren’t. It would be nice if Microsoft would sign everything in the OS so that a single certificate rule—rather than a less-secure path rule—would allow everything to run. In the meantime, you’re forced to rely on an insecure path rule or come up with hash rules for everything.

A certificate rule works similarly, by identifying a particular certificate and allowing any software signed by that certificate. You could, for example, use a single code-signing certificate to sign all in-house applications, then define a single rule that would allow them all to run under SRP.

This technique has a side benefit: carefully control access to the certificate and you can prevent new, unauthorized versions of the software from running, too. It’s an effective way to help control change in your environment.

The way to use SRP, then, is to define rules to identify all allowed software (and again, I’m not suggesting that’s a one-day task—I know how daunting it is, but this is security, folks), and disallow everything else.
**Microsoft .NET Framework Security**

The .NET Framework, as a totally new way of writing and running software, fixes many of the security problems existing in traditional software. Because all .NET software runs inside the .NET Common Language Runtime (CLR), the CLR provides a single point of security for deciding what will and will not be allowed to run. You can configure this “runtime security policy” to have code groups and permission sets. A code group is essentially a set of rules—not unlike those used in SRP—that identify software. A permission set is a set of permissions that is assigned to a code group. Code groups are evaluated dynamically each time software is executed by the CLR; the first code group a piece of software fits into determines that software’s permissions. By default, only an “All Code” group exists, which is the default group for any software not falling into any other code group. However, you can define your own code groups.

For example, in Figure 7.16, I’m defining a new code group that will identify software based on the publisher that issued the certificate used to sign the file. I can also identify software based on a hash, its path, or a number of other criteria, not unlike SRP.

![Create Code Group](image)

**Figure 7.16: Creating a new .NET Code Group.**

Having identified software with one or more code groups, I can create permission sets that define what software is allowed to do. Permission sets are then assigned to code groups. Several permission sets exist by default; you can create your own and they can be detailed. For example, in Figure 7.17, I’ve selected the EventLog permission for a new permission set.
Figure 7.17: Defining a new permission set.

Figure 7.18 shows the details of this permission, which provides Browse permission to the event logs on server2. That’s detailed. Code groups assigned this permission set will have this, and no other, permissions. You can see that the list of permissions you can define is extensive, including file access, AD access, dialog boxes, message queues, databases, general security, and tons more. Each can be configured with detailed settings determining what applications can do.
Like SRP, you can prevent all unauthorized .NET software simply by assigning the built-in “Nothing” permission set to the “All_Code” code group. Any code not falling into one of your code groups winds up in All_Code by default; receiving “Nothing” permissions means the code won’t even run.

The really cool part about the runtime security policy is that it’s available on all machines running .NET—which means it works in Windows 95 and later. Although it only affects .NET code—and that code is still somewhat less common than traditional code—it’s a great idea to get on top of your .NET security. .NET viruses are certain to start appearing eventually; if you’re identifying your .NET software (easy to do now that there’s less of it in most environments), you can shut down all unknown software right from the get-go.

Figure 7.18: Details of a permission set entry.
**Windows Script Host TrustPolicy**

Windows scripting—VBScript, JScript, and so forth—has a pretty harsh reputation when it comes to security. Many script-based viruses have been distributed, and many environments seek to simply disable scripting altogether. Doing so is difficult—the WScript.exe and CScript.exe files that host a script are protected by Windows File Protection; simply deleting them is ineffective. The files are also replaced by service packs. Reassigning filename extension associations is poor protection, too, because both executables will execute any script file passed in a command-line argument, regardless of extension. However, Windows Script Host (WSH) 5.6—the current version installed by any recent service pack, and downloadable from the Microsoft Web site—includes a built-in TrustPolicy system.

Windows scripts can be digitally signed, just like most software. If the digital certificate used to sign a script was issued by a trusted publisher, a script is considered trusted. WSH can be configured to execute only trusted scripts, giving you the opportunity to write and run your own scripts but to stop script-based viruses fairly easily. It’s all configured through a series of registry keys and values:

- **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows Script Host\Settings**
  - UseWINSAFER—Set to 1 by default, this turns off the TrustPolicy system in favor of SRP; set this to zero to enable TrustPolicy
  - TrustPolicy—Set to 0 (the default) to run all scripts, 1 to prompt the user for untrusted scripts (poor protection because most users will just allow the script to execute anyway), or 2 to execute only trusted scripts
  - SilentTerminate—Set to 0 to not display anything when a script can’t be executed because it’s untrusted; set to 1 to display an error dialog box

- **HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows Script Host\Settings**
  - TrustPolicy—Same settings as above. If configured per-user, this setting will override the machine-wide setting

WSH TrustPolicy is an effective means of filtering and controlling scripts in your environment.

**Summary**

This chapter covered a lot of ground—all related to software management. We started with an exploration of software maintenance, looking at ways in which the boring job of patch management could be made a bit easier and more effective. I also discussed software filtering and showed you ways in which the software allowed to execute in your environment could be locked down a bit more. All of these techniques can serve to make any Windows environment more secure, and they can help reduce the management overhead involved with securing a Windows environment.

The next chapter will cover network security. We’ll look at various tools that can help create a more secure environment and help prevent and detect attacks as well as techniques for making your network more secure. In fact, the next chapter will discuss in much greater detail the multi-layered defense and network segmentation technique that this chapter touched on. We’ll also look at network technologies—such as 802.1X—that can provide a more secure network.
Chapter 8: Securing the Network

Of course, the network—the backbone upon which the enterprise functions—is a major source of potential security problems in a Windows—or any other—environment. Securing the network is a critical requirement in order for the overall enterprise to be secure, so this final chapter will focus on often-overlooked network security problems and solutions.

Security Through Architecture

Many techniques exist for securing the network. Some of the common techniques are probably in use on almost every corporate network in the world:

- Use firewalls to protect the network from Internet-based attacks
- Use common wireless security mechanisms, such as Wi-Fi Protected Access (WPA), to secure wireless connections
- Use authentication and authorization to protect network-attached resources, such as file servers, from unauthorized access

Unfortunately, one area that can provide excellent security but is commonly overlooked is securing network architecture. The reason more secure architecture techniques are often overlooked is that the intranet is often seen as homogenous when it comes to trust, access, and security. In other words, once you’re in the intranet, you can do whatever you want. Of course, nothing could (or at least should) be further from the truth; most security problems come from within the intranet, completely unaffected by the firewalls and other technologies meant to protect the network from attack. It’s these “inside jobs” that can be prevented by better security architecture.

Resource Clustering

A good practice is to start an architectural examination of the network by clustering resources, at least logically if not physically. As Figure 8.1 shows, this process involves grouping resources that have common, or at least similar, security and communications requirements. For example, file servers rarely need to communicate with one another but always need to communicate with client computers. They have common security needs—perhaps authenticating users to an Active Directory (AD) domain—and are often administered in the same way. Extranet Web servers need to be accessible to select Internet users (usually company business partners of some kind), and may also need to be accessible to intranet users. Figure 8.1 shows these resources being grouped into common network segments or subnets—a first step toward a more secure architecture.

A network _segment_ is usually defined as a group of network hosts connected by a common medium or device, such as all devices connected to a particular hub. However, with the prevalent use of switches and VLANs, a _subnet_—a contiguous IP address space—is more useful for breaking the network apart. For example, all file servers in the network might be attached to a single subnet and share a contiguous address space, even though they might be located in separate sections of a building or (in particularly creative VLAN architecture) across WAN links.
Chapter 8

Figure 8.1: Grouping common resources together.

The primary reason behind this resource grouping is to get resources with common security and communication requirements into common, independent address spaces. The reason to do so is that network security is relatively easy to manage on a per-subnet basis, through segmentation, which we’ll explore in the next section. Figure 8.2 shows a more detailed, real-world example of resource grouping, in which resources are more likely to be sub-grouped by access requirements.
Figure 8.2: Grouping resources by access requirements.

This example only deals with file servers, which is typically a large number of servers and other resources in an organization. Here, the resource grouping of file servers has been further broken down into those accessed by Human Resources (HR), company executives, general company users, and a sales department’s users. This breakdown is important because of their unique security requirements and communications needs. For example, sales department employees might have no need to communicate with HR file servers, making the separation between these servers useful.

Does each sub-grouping need a distinct IP address subnet? Perhaps. Certainly, as you’ll see in the next section, doing so can provide for more granular security. Besides, as most organizations use private IP addresses on their intranets, there is no shortage of IP addresses to work with, so why not break down the network into smaller, manageable chunks?
**Segmentation**

Once your resources are grouped into distinct subnets, you can begin the process of *segmentation*, which means restricting access to each group’s independent subnet based on the security and communication needs of that group. For example, consider the network diagram in Figure 8.3.

This example shows departmental subnets (which might include file servers and other department-specific servers) connected to a common Clients subnet. Additional subnets provide access to resources such as an extranet and email, which are shared across the organization rather than being specific to a department. The important feature here is that the subnets are connected by firewalls.

> Of course, you don’t *need* to use actual firewalls; any device capable of providing basic firewall-like functionality—such as a router—can get the job done.

*Figure 8.3: Segmenting the network by using firewalls.*
The intent in this example is to acknowledge two important facts:

- Not all client computers need to communicate with all server resources
- The fewer servers a client can communicate with, the less damage it can do in the event that it becomes a platform for attack

Although Figure 8.3 shows the Clients segment as one contiguous subnet, in reality, you could use additional firewalls to break down client computers into individual subnets, perhaps grouping a couple of dozen or so clients per subnet.

The firewalls between the subnets would then be used to restrict communications between the subnets. For example, client computers sitting on an HR Clients subnet might be the only ones permitted to communicate with the HR Servers subnet, and even those communications might be restricted to the selection of UDP and TCP ports actually used for authorized communications.

Does this architectural model create additional administrative overhead? Absolutely. Does it serve any purpose in increasing security? Definitely. A problem that the IT industry currently faces is the myth that the intranet can be trusted and therefore allowed to communicate however and whenever it wants with whatever other intranet resources. It is in this manner that viruses spread (and how other internal problems, such as embezzlement, occur). For example, when the Code Red worm was released for IIS 5.0, it quickly spread throughout networks because it was allowed to do so. Certainly, Microsoft’s decision to install IIS by default on all Windows 2000 (Win2K) computers contributed to the problem, but had those computers had more restricted communications between one another, the problem would have been stopped cold. For example, do client computers in your company typically need to communicate with one another via HTTP? That is rarely the case, so why allow the communication? After all, HTTP communication was a big part of how Code Red was so successful at replicating itself.

The IT industry has long realized that the Internet is not to be trusted, and has taken steps to secure communications with it. Similarly, you need to realize that your own intranet can’t be fully trusted and you must take steps to secure internal communications to help restrict the effect of an internal attack.

**Creating Trust Zones**

Another way to help secure internal network links is to stop considering the entire intranet to be a uniformly trustworthy area, and instead to define intranet trust zones. Figure 8.4 shows an example. Here, trust is based primarily on your ability to secure physical connections. The corporate LAN segment, for example, represents shared resources such as file servers and email servers. Clients connected to the inside office—that is, using network wiring only accessible from within a physically secured area—are granted the most trust because you can exercise more control over who connects to this network. The firewall allowing access to the corporate LAN might be the most permissive.
In contrast, conference room connections—which can be used by non-employees using computers they’ve brought from who knows where—are inherently less trustworthy. Although some access to the corporate LAN might be permitted, such access might be more restricted—simply providing access to a few less-sensitive file servers, for example, as well as access to the Internet.

Although not illustrated in Figure 8.4, the inside office connections would likely be allowed to connect to the Internet, as well, through an appropriate firewall.

Wired connections in the building lobby, which offer the most opportunities for unauthorized individuals, might be provided only with Internet connectivity, allowing visitors to check their email, for example, but providing no connectivity to the resources on the corporate LAN.

This entire trust model is based on your ability to physically control the computers connecting to these various areas. The less physical control you have—to keep patches and antivirus software updated, for example—the less you trust the computer, and the less you trust the zone in which it can connect.

What about wireless connections, which aren’t tied to a physical area? Base your trust of wireless connections on the difficulty an attacker would have in connecting. Fully secured wireless connections that use WPA or other reliable security protocols might be considered fully trusted; less-secure or completely open wireless networks (perhaps ones provided for visitors’ use) would be untrusted.
Chapter 8

The Architectural Big Picture

This concept of trusted zones becomes an overlay on your segmented, resource-grouped security model. Thus, not only do HR clients have exclusive connectivity to HR servers, but those clients must be connecting from a trusted zone; untrusted zones don’t provide any connectivity to the IP address ranges used by HR servers. This concept is very different than simply allowing free-flowing connectivity and restricting resource access based on authentication to AD. Of course, you’re still restricting access based on NTFS (or other appropriate) permissions, but you’re also preventing unauthorized client computers from even establishing a connection to these resources.

Figure 8.5 shows an example of the big picture. Complicated? It certainly is. But security is complicated and you shouldn’t try to artificially simplify it. In reality, a complex communications plan such as this—which shows exactly which client segments are allowed to communicate with which server segments—might be implemented as a large set of rules within a single, centralized router or firewall. Once established, relatively little ongoing maintenance would be required. Notice, too, that server segments don’t communicate with one another, other than as required.

Not all connections are shown; obviously, subnets would need to communicate with a domain controller subnet, and the Internet connectivity also isn’t shown. But hopefully this simplified diagram communicates the overall concept of segmenting communications.

Figure 8.5: Creating a more secure communications plan.
What about the inevitable user who needs to access multiple subnets, such as an HR Director who needs access to both the HR and Exec subnets? You have a few options. You could simply bind two IP addresses to the person’s computer, making them part of both the HR Clients and Exec Clients subnet (which are each likely to be a VLAN, anyway). You could enter the correct firewall rules for that one client. You could create a new client subnet that has access to both server subnets (the networking equivalent of creating a new user group to represent the unique access needs). This makes network administration as complex as user group administration—and shouldn’t it be?

Remember, the overall intention is to acknowledge that clients who will never have permissions to access a computer shouldn’t be able to connect to that computer. This philosophy accommodates the fact that certain anonymous connections (such as HTTP) can be leveraged for malicious purposes, and that vulnerabilities in the Windows OS can turn a simple connection into a major exploit. Although patch management, antivirus scanners, and other techniques can help mitigate these potential problems, an additional layer of defense—simply denying connectivity whenever possible—helps seal the deal and make the network that much more secure.

Security Through Monitoring

Monitoring is another critical aspect of network security, as it allows you to ensure that the network you designed and implemented stays the way you intended. It also allows you to detect new situations and circumstances related to security and to respond to them appropriately. Unfortunately, most network administrators do little or no routine monitoring, which creates significant opportunities for attackers.

Quarantine

One rapidly emerging concept in network security is the idea of quarantine. Although not yet implemented in a significant fashion within Windows, it’s expected to be a major component of the forthcoming Windows Longhorn release.

The purpose behind quarantine is actually similar to the purpose behind trust zones, which were described earlier. Connections from your building’s lobby, for example, are untrusted because you have no idea what is going to be plugged into there, and you have no control over it. Of course, location isn’t really a definitive factor; a laptop plugged into your inside office network can be just as untrustworthy, especially if it’s been off the network—and outside your control—for a significant period of time. Quarantine seeks to create a more dependable indication of trust.

Essentially, quarantine consists of both a client and server component. The client component is responsible for various scanning and inventory tasks, not unlike a systems management client of the kind installed by Microsoft Systems Management Server (SMS) or similar software. The server component contains network access policies. Those policies dictate certain requirements for accessing various different network resources. As Figure 8.6 shows, if a client can’t meet the minimum requirements, the client isn’t permitted to access those resources.
Here’s an example: Full access is allowed to patch management servers, without restriction, because clients will need access to those servers in order to meet other requirements (this is a typical quarantine scenario—let clients get enough access to fix whatever problems they have). In order to access the Internet, clients must have a personal firewall installed and enabled (updated antivirus software might be another requirement). In this example, a personal firewall hasn’t been detected, so Internet access isn’t allowed. Access to corporate resources might require authentication, up-to-date patches, and up-to-date antivirus software; in this example, one of those requirements hasn’t been met, so access is denied.

The purpose behind these policies is to guarantee a certain level of end-to-end security within your network. Clients without up-to-date patches might have been the victim of an exploited security vulnerability, and are denied access to resources that might be the target of the exploit’s attack. Because infrastructure devices—such as routers and switches—will participate in quarantine (Microsoft has done extensive work with Cisco in creating standards for this purpose), computers that don’t meet minimum requirements will have their connectivity denied, almost as if a smart firewall were in place. Thus, you’ll be able to define what you consider a secure client to be, and ensure that only secure clients are allowed to connect to sensitive resources.
Network Sniffing

The value of a network sniffer—or packet capture tool, if you prefer the formal term—as a security tool is somewhat dubious. Although it’s technically possible for you to watch packets fly by and detect anomalous activity, it’s not likely to happen. In fact, network sniffers are more likely to be used as part of an attack rather than as a means of detecting one. For this reason, Microsoft’s own network sniffing tool—Network Monitor, a version of which is included with Windows and a more flexible version of which is included with SMS—has its own security precaution built into it (see Figure 8.7).

Microsoft realized that Network Monitor (or NetMon, for short) could be used to attempt to capture sensitive information from the network as part of an attack. For this reason, every copy of NetMon periodically transmits a special packet in the Bone protocol. Other copies of NetMon can capture and identify this packet, telling you that a copy of NetMon is in use, and giving you the MAC address of the computer that’s running it.

Microsoft’s internal code-name for NetMon is Bloodhound, which makes the name of the Bone protocol a sort of in-joke.

Periodically running a copy of NetMon and looking for Bone packets from computers other than your own will tell you whether NetMon is in use on your network. If it is, you can take appropriate steps. However, there are some tricks to ensuring you actually see the Bone packets that are transmitted:

- Leave a copy of NetMon running continuously to catch any packets. Have NetMon use a large capture buffer and drop packets other than those you’re specifically looking for.
- Because switches can prevent your workstation from seeing all the traffic on the network, consider running NetMon on a standalone workstation that is connected to a promiscuous switch port—one that will echo all traffic that passes through the switch. Doing so helps to ensure that any transmitted Bone packets are, in fact, received.
Figure 8.7: Using Network Monitor as a security tool.

Of course, NetMon is the only such tool that uses the Bone protocol, so other network sniffers won’t be detected this way. A better technique would be to use Software Restriction Policies to define the software you want to allow on your network, and to prevent other software—including network sniffers—from running. Still, because a non-domain computer wouldn’t be affected by Software Restriction Policies, monitoring for the Bone protocol can still help spot attackers who are using the readily available NetMon tool.
Port Monitoring

Port monitoring is an absolutely essential part of ongoing network security maintenance. It doesn’t need to be difficult, either; port monitors (or scanners) are easy to come by. What you should have is a complete list of every computer on your network and the ports that are allowed to be open on each (for client computers, this might be as simple as saying “no ports should be open”). You can then routinely scan computers to ensure that only authorized ports are open. Figure 8.8 shows a simple, free port scanner available from http://www.angryziber.com/ipskin/.

![Angry IP Scanner](http://www.angryziber.com/ipskin/)

**Figure 8.8: Angry IP Scanner.**

As you can see, the scanner can automatically scan your entire network—based on IP address ranges—and report on each host it finds, listing the open ports on each. Simply compare this list with your list—especially on server computers—and take whatever action is necessary if unauthorized ports are found to be open.

As a bonus, this particular IP port scanner has an add-in that can display Windows shared folders, which are another excellent thing to monitor from a security standpoint, and the tool makes it easier to see which shared folders exist on each scanned computer.
Hundreds of other port scanners exist. In fact, Windows can easily display open ports simply by running

```
netstat -a -o
```

from a command-line. As Figure 8.9 shows, you’ll not only get a list of ports but also the process ID (PID) of the process that has the port open, making it easier to determine which software opened the port (use Task Manager to look up the PID and see the process name). Although Netstat can’t be used to scan ports on a remote computer, it is an effective way to examine the ports opened on a local computer and figure out what software has the port open.

You can also run Netstat –a –b to display the executable name associated with each open port.

![Figure 8.9: Using Netstat to monitor open ports.](image)

Only *listening* ports are ones you need to be worried about; established ports may be used for *outgoing* communications, which aren’t typically a concern.
Lockdown Tools

Locking down ports on Windows computers can be tricky, especially on servers. One wrong move and you can completely disable critical network services. The primary way in which ports are locked down is to disable the software that opened the port. You can also use local firewall software to simply block the port, whether a piece of software wants it open or not.

With Windows Server 2003 (WS2K3) Service Pack 1 (SP1), Microsoft introduced the Security Configuration Wizard, a tool that creates security templates to help achieve server lockdown. This tool includes an XML-formatted database that allows the wizard to understand the complex inter-service dependencies and port requirements in a Windows server, making it possible for administrators to simply indicate which roles a server is fulfilling—domain controller, file server, and so forth—and have the wizard create a security template that, when applied, will lock down the server correctly.

The wizard’s templates can even provide proactive security, ensuring that any additional services that might be installed won’t be able to start—and open any ports—until you modify the template (by using the wizard) to specifically allow that service.

Some of the commonly used ports you might find on a Windows computer include:

- TCP 20-21—FTP
- TCP 22—SSH
- TCP 23—Telnet
- TCP 25—SMTP
- TCP/UDP 53—DNS
- UDP 67-68—DHCP
- UDB 69—TFTP
- TCP 80—HTTP
- TCP 88—Kerberos
- TCP 110—POP3
- TCP 119—NNTP
- UDB 123—NTP
- TCP 139—NetBIOS Session
- TCP 143—IMAP4
- TCP 389—LDAP
- TCP 443—HTTPS
- TCP 445—Microsoft DS
- UDG 445—Microsoft SMB
- TCP 636—LDAP over SSL
- TCP 993—IMAP4 over SSL
- TCP 995—POP3 over SSL
• TCP 1344, 1434 (also on UDP)—SQL Server
• TCP 1494—Citrix MetaFrame
• TCP 1863—Windows/MSN Messenger
• TCP 3389—Terminal Services/RDP
• TCP 5190—AOL Instant Messenger (AIM)
• TCP 5222, 5223, 5269—Jabber
• TCP 5800, 5900—VNC

You can find more comprehensive lists at http://en.wikipedia.org/wiki/List_of_well-known_ports_(computing) and http://www.iana.org/assignments/port-numbers. The latter address is the official list of registered ports, and it includes port assignments through 49,151 (although not every port is assigned). The maximum port number is 65,535.

Many Microsoft products use endpoint mapping. Exchange Server is an excellent example—clients contact Exchange on a well-known port, connecting to the server’s Remote Procedure Call (RPC) service. Exchange then dynamically opens a new port (usually well above 10,000) and assigns the client to that port. This technique allows Exchange to spread clients across ports, which provides better performance but makes it difficult to tell which ports are legitimate and which aren’t.

Exchange—and other applications—often allow you to configure them so that they use a very narrow range of ports, making it easier to work with them. However, this generally degrades the application’s performance as well.

**Intrusion Detection/Prevention Systems**

Intrusion Detection Systems (IDSs) and Intrusion Prevention Systems (IPSs) act as, to one degree or another, automated monitoring and response systems. Essentially, they continually monitor your network for suspicious activity, then do something about it. An IDS is more likely to alert you upon detecting something amiss, while an IPS may be programmed to take some corrective or defensive action. A Common Intrusion Detection Framework (CIDF) defines four components for IDSs:

- “A” boxes, which analyze network activity
- “C” boxes, which include countermeasure mechanisms or response procedure equipment
- “D” boxes, which are disk storage mechanisms—essentially, logging devices
- “E” boxes, which are event generators, or sensors
Obviously, a single device or software application can encompass one or more of these pieces of functionality; a commercial IDS or IPS generally includes all four (with the “C” portion being what sets an IPS apart from an IDS, because the countermeasure mechanism is generally some sort of active response).

Different systems use different means of detecting potentially unwanted activity:

- **Suspicious detection** is used to flag a particular activity, such as Telnet usage on a network on which Telnet isn’t normally used. Port scans are another suspicious activity.
- **Abnormal detection** is used to look for abnormal behavior, such as after-hours access to a particular resource.
- **Signature or pattern detection** looks for specific patterns of activity in much the same way an antivirus solution looks for patterns of code. A database usually defines a set of patterns known to be associated with particular types of attacks. This can, for example, be used to spot a worm that is transmitting itself across the network.

Better systems will obviously incorporate two or all of these techniques to provide the best coverage. In terms of form factor, IDSs are common in both hardware and software formats. For example, Figure 8.10 shows a partially disassembled Cisco WS-X6381-IDS unit; a hardware “black box” that is connected to your network and administered remotely.

![Figure 8.10: Cisco hardware IDS.](image)

Figure 8.11 shows a software IDS, the Windows-based KFSensor. This application scans for a variety of different potential attacks and can alert you to suspicious activity. As shown, the application has detected activity that appears to be an IIS-based worm attempting to propagate itself.
Chapter 8

Figure 8.11: Software IDS.

This system uses signature matching to detect the worm. This sort of system can be a valuable last resort in your overall security scheme, because it will let you know that the attack is taking place even if all of your active defenses fail. You can then act immediately to stop the attack, or mitigate its effects, before it gets completely out of control. A system such as KFSensor provides a complete signature database (see Figure 8.12) to detect attacks. Of course, just like an antivirus solution, this database must be continually updated to remain effective against emerging attacks.
Chapter 8

Figure 8.12: Signature database in KFSensor.

You should be able to define your own signatures in an IDS (KFSensor allows you to), as well as import signatures from the popular SNORT format (KFSensor allows this, too).

One key way that IDSs detect attacks is by acting as traps, or *honeypots*. With this technique, the IDS doesn’t continually scan all network traffic—which, depending on the size and configuration of your network, might be impractical or difficult to set up—but rather sets itself up to be attacked. The IDS simulates various services and OSs, essentially appearing to would-be attackers as a wide-open server. By placing various IDS *sensors* throughout your network, you can be sure of attracting attacks and spotting them before they get too far on your network.

For example, Figure 8.13 shows another software IDS, Specter Control, being used to emulate a Mac OS X computer with open FTP, Telnet, SMTP, Finger, HTTP, and POP3 ports. It’s providing traps for DNS, IMAP4, SUN-RPC, SSH, and generic traffic. By emulating an “open” system, the IDS will be one of the first attack targets (especially if your other hosts are locked down). Once attacked, the IDS can provide full details about where the attack is coming from, allowing you to stop it.

A really flexible IDS system will combine passive network scanning—along the lines of network sniffing—with honeypot techniques. This feature allows the IDS to see attacks being made against other systems as well as provide a target for attacks. Because both types of monitoring—sniffing and honeypot—are useful, you might use different IDS systems that each provides one of these techniques.

Keep in mind that a sniffer-style IDS is limited to the traffic you allow it to see. For that reason, they’re usually connected to switch ports that have been defined as *promiscuous*, allowing the IDS to “see” all of the traffic passing through the switch.
IDS deployment techniques often suggest placing a honeypot IDS in your perimeter network (such as alongside your public Web servers) to attract attackers. This method is a good idea, but it presumes that all attacks will come from outside your network, which isn’t the case. Deploy IDSs everywhere an attack might originate or target, including throughout your network and most especially alongside resource servers, internal Web servers, and even on a handful of client segments. Many, many, many modern attacks first target more vulnerable client computers, often through social engineering (sending in a virus disguised as an animated postcard from a family member, for example). Once established on a trusted computer, the virus then launches more subtle attacks against corporate resources, including servers. Always remember that your internal network can never be completely trustworthy; treat it in much the same way you would the Internet by minimizing the exposure corporate resources have even to internally launched attacks.

Figure 8.13: IDSs emulate open systems to attract attackers.
Security Through Technology

There are a number of technologies that can, in and of themselves, help to increase the security on your network. Some are a bit complicated to deploy and may require careful planning, but they’re unique in that, no matter what type of network you’re operating or what your security issues are or business needs may be, these can almost always provide an “instant win” by helping increase the level of security you’re working with.

802.1X

Although 802.1X is often associated with wireless security (an area I’ll touch on in the next section), it is in fact equally applicable to wired networks. 802.1X is an IEEE standard for port-based network access control, requiring devices—rather than users—to authenticate before they’re actually connected to the network. 802.1X requires devices—such as routers, client computers, and other connected equipment—to have a client component, while network switches typically provide the access control point. Typically, 802.1X uses the Extensible Authentication Protocol (RFC 2284) to handle authentication.

When fully implemented, 802.1X can be a thing of beauty. No longer do you have to worry about trusted and untrusted zones of your network; only trusted clients—those who can authenticate—can obtain network “dial tone.” Unauthorized computers can’t even get an IP address because they’re not granted network access. Network switches usually rely on RADIUS for authentication, as depicted in Figure 8.14.

![Figure 8.14: Using RADIUS in an 802.1X environment.](image)
Here, the client—known as a supplicant in 802.1X terminology—passes authentication credentials to the switch. The switch then passes these to a RADIUS server, which may pass them on to another directory—such as AD—for processing. When the RADIUS server gives the switch a positive response, the switch begins transmitting network traffic for that client. Without the positive authentication, however, the switch port the client is connected to is never fully activated, cutting the client off from the entire network.

Windows has supported 802.1X since Windows XP and WS2K3, although to date, implementations have primarily focused on wireless networks, where it’s far easier for unauthorized individuals to “plug in” to the network. However, deploying 802.1X for your wired networks will eliminate the possibility of an attacker compromising your physical network—accessing the network from the lobby or other public locations—and increase the overall level of security on your network.

**Wireless Security**

In addition to using 802.1X to secure actual connections to your wireless networks, using a wireless security protocol can help protect the contents of wireless transmissions and keep unauthorized users off the network.

Originally, the Wired Equivalent Privacy (WEP) protocol was intended to provide the same level of privacy one could expect with a wired connection (which isn’t actually all that much privacy). However, the protocol was quickly cracked due to some inherent vulnerabilities, and the industry as a whole became wary of it. WEP has since been superseded by WPA, which has been codified in IEEE standard 802.11i (called WPA2). WEP continues to provide a minimal level of security, but WPA is considered the minimum level of security for sensibly run wireless networks.

802.11i utilizes the Advanced Encryption Standard (AES) block cipher for encryption (WEP and the original WPA only used a less-secure RC4 stream cipher). 802.11i actually requires 802.1X for port-level authentication. 802.11i is designed to use 802.1X for authentication, or it can use pre-shared (symmetric) keys. Pre-shared keys are easier to implement (because you essentially just configure the wireless access point—AP—and your client devices with a password) but is considered less secure because the password is rarely (if ever) changed and because it’s used too much. Windows XP supports WPA2/802.11i in the latest service pack, although you may also need updated drivers for your wireless network adapter in order to take advantage of the service.
**Secure Network Adapters**

Major network adapter manufacturers, including companies such as Intel and Broadcom, are beginning to incorporate security features into network hardware. Typically, these products are developed with unique embedded digital certificates that uniquely identify the network adapter. Because these certificates cannot be forged (the certificates themselves are often digitally signed using the adapter manufacturer’s own private key), they represent a more secure form of identification than MAC addresses (which can easily be forged). Network infrastructure devices—such as DHCP servers, switches, and so forth—can be designed to identify devices by their unique certificate, helping to ensure that only authorized, known devices are allowed to access the network and its services. The embedded certificate can also be used to speed network-level encryption and other services.

**IP Security**

IP Security (IPSec) is a broad suite of protocols designed to improve network security. From an implementation standpoint, IPSec can be viewed as a set of filters and actions. For example, a filter might define all traffic destined to a particular server, or all traffic sent over a particular protocol. When traffic matches a filter, a companion action is applied to that traffic. Actions may be as simple as dropping the traffic (thus giving IPSec firewall-like capabilities), or it might require the traffic to be encrypted or authenticated. Figure 8.15 illustrates this process.

*Figure 8.15: IPSec filters and actions.*
For example, you might define a filter for all ICMP traffic and another for all IP traffic. Figure 8.16 illustrates how Windows XP might list two such filters.

![Figure 8.16: IPSec filters in Windows XP.](image)

You might define rules that drop traffic, allow it, request security (but make it optional), or require security. Figure 8.17 lists these rules in the Windows XP IPSec interface.
Figure 8.17: IPSec rules in Windows XP.

An IP Security Policy (or IPSec Policy) joins filters and actions: For example, you might create a policy that drops ICMP traffic or one that requests security for all other IP traffic.

IPSec obviously requires a lot of planning before implementing it across your organization, but it’s yet another way in which your organization’s network security can immediately benefit a great deal. IPSec can be centrally controlled through AD Group Policy, allowing you to create centralized IPSec policies for your entire organization. This setup can help ensure that sensitive data is never transmitted in clear text; when combined with technologies such as 802.1X to prevent unauthorized network connections, IPSec can act as part of a layered defense that makes your network as close to absolutely secure as possible.
Summary
This chapter covers several techniques, technologies, and tools that can be used to help close the loop on Windows enterprise security. By providing a better, more secure network backbone on which Windows can operate, you’ll help close security loopholes, defend against common attacks on Windows systems, and provide an overall more secure environment for your users.

Windows security is a matter of details: Paying attention not only to major security issues but also the many commonly overlooked issues this guide has explored will help you maintain a more secure, more reliable Windows network. Security has become the lynchpin for a number of enterprise needs, including compliance, reliability, accountability, and more; by creating a more secure Windows enterprise, you’ll be helping your organization meet a number of important business needs. Good luck!

Content Central
Content Central is your complete source for IT learning. Whether you need the most current information for managing your Windows enterprise, implementing security measures on your network, learning about new development tools for Windows and Linux, or deploying new enterprise software solutions, Content Central offers the latest instruction on the topics that are most important to the IT professional. Browse our extensive collection of eBooks and video guides and start building your own personal IT library today!

Download Additional eBooks!
If you found this eBook to be informative, then please visit Content Central and download other eBooks on this topic. If you are not already a registered user of Content Central, please take a moment to register in order to gain free access to other great IT eBooks and video guides. Please visit: http://www.realtimepublishers.com/contentcentral/.