

realtimepublishers.com[®]

The Definitive Guide[™] To

Securing Windows in the Enterprise

SCRIPTLOGIC

Don Jones

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.

Introduction to Realtimepublishers.....	i
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	24

Copyright Statement

© 2005 Realtimedpublishers.com, Inc. All rights reserved. This site contains materials that have been created, developed, or commissioned by, and published with the permission of, Realtimedpublishers.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 Realtimedpublishers.com, Inc or its web site sponsors. In no event shall Realtimedpublishers.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.

Realtimedpublishers.com and the Realtimedpublishers 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 Realtimedpublishers.com, please contact us via e-mail at info@realtimedpublishers.com.

[**Editor's Note:** This eBook chapter was downloaded from Content Central. To download other chapters from this eBook, please visit <http://cc.realtimedpublishers.com/portal.aspx?pubid=335>.]

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 polities 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.

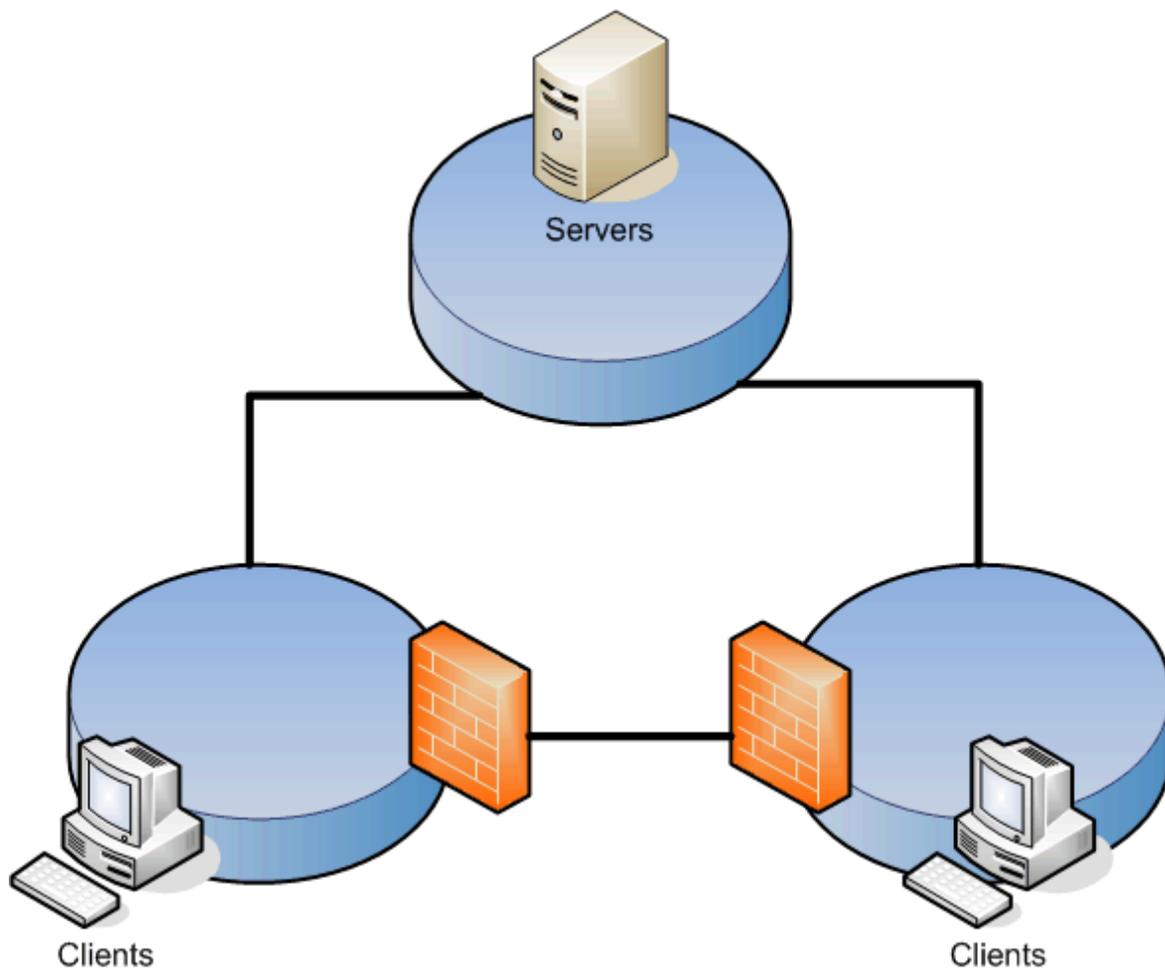


Figure 1.1: Using firewalls to compartmentalize a network.

 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.

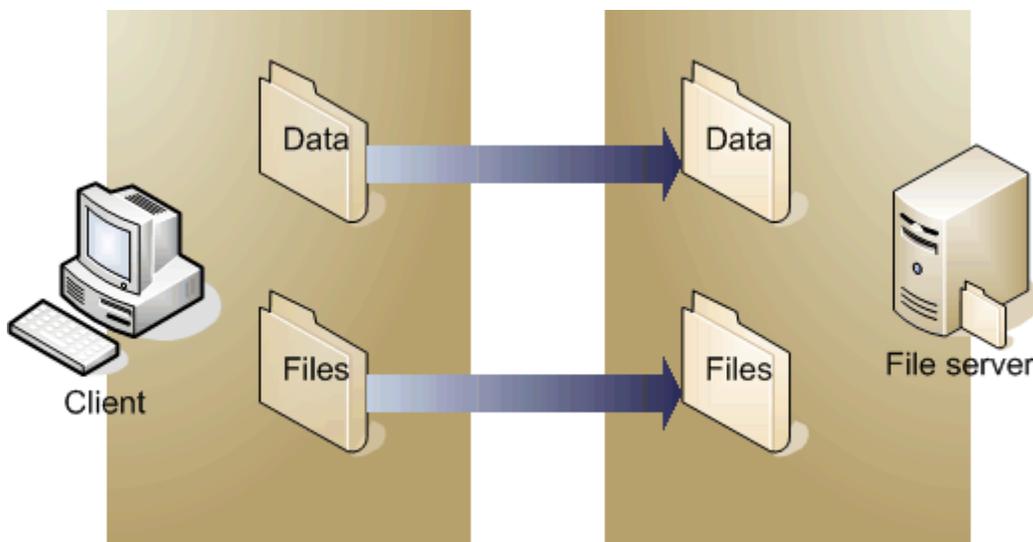


Figure 1.2: Redirecting local folders provides users with convenience while actually keeping files in a more readily secured location.

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.

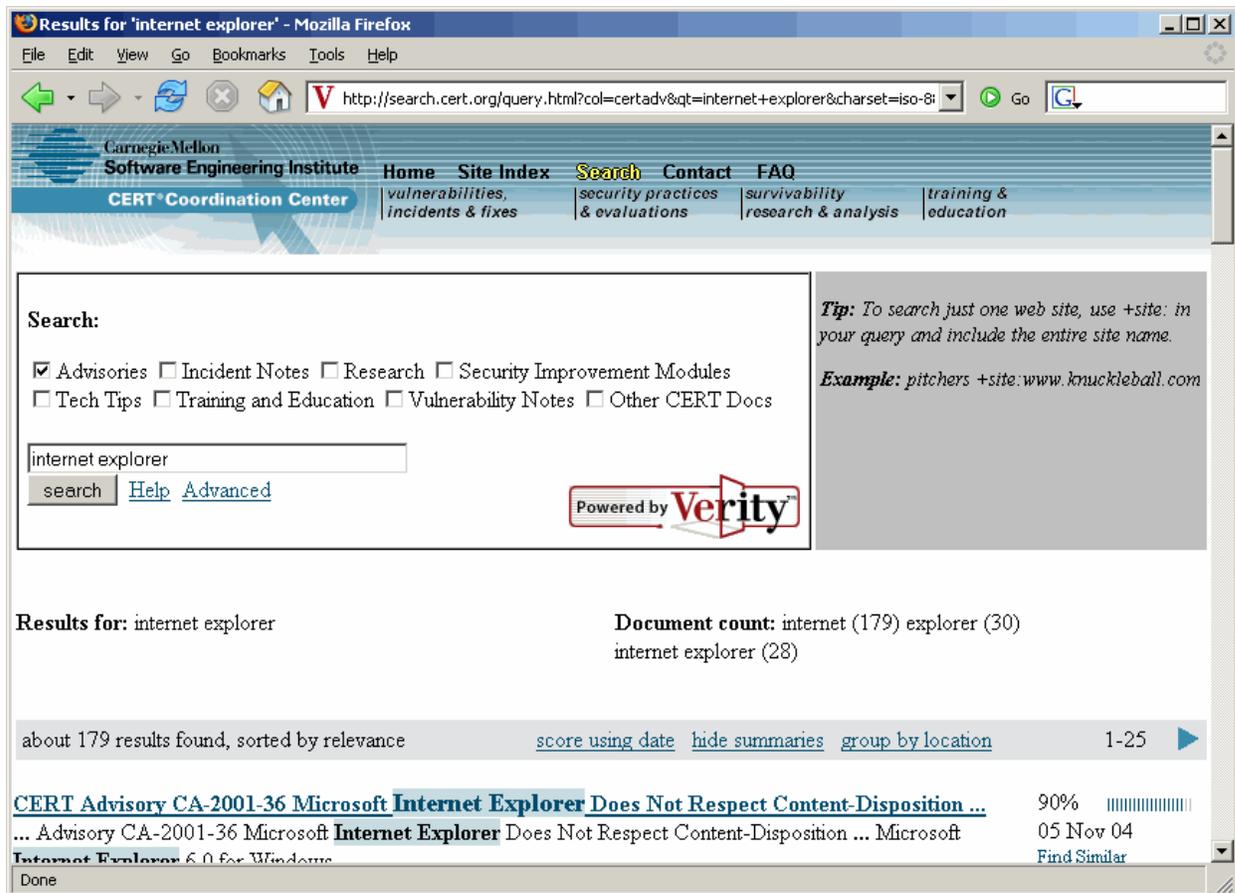


Figure 1.3: Reviewing IE vulnerabilities at the CERT Web site.

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.

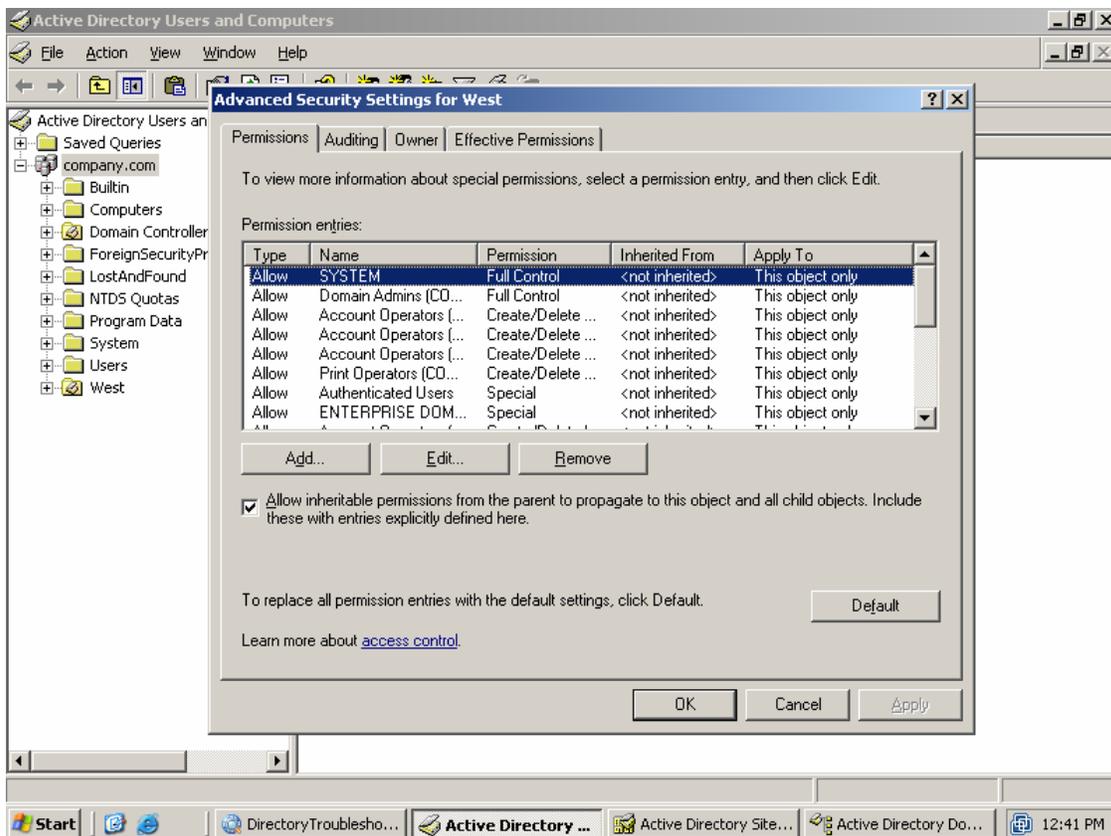


Figure 1.4: Reviewing permissions in AD Windows' built-in user interface.

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.

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.



Traditionally, 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.

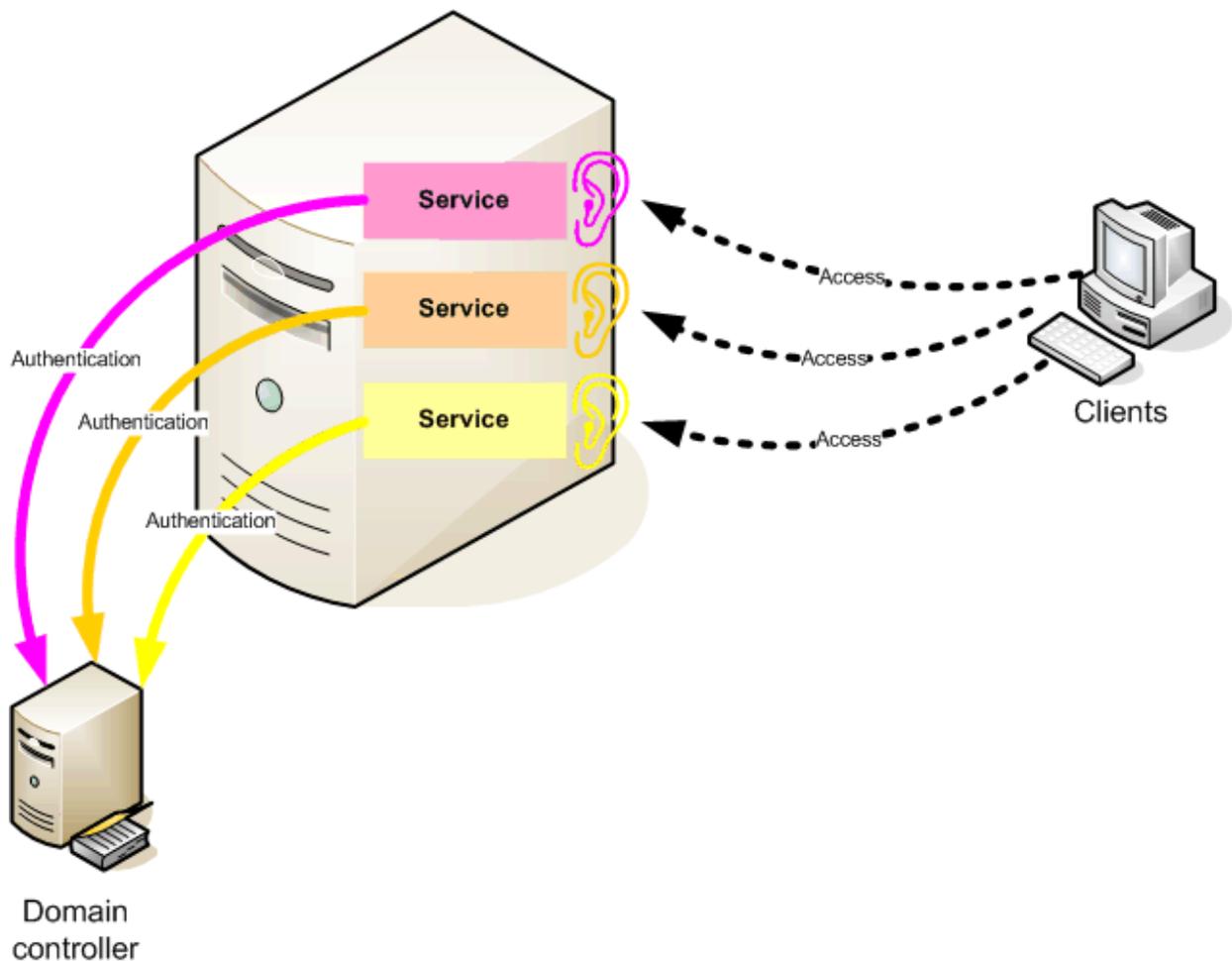


Figure 1.5: Services present a dual security issue through access and authentication.

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.

 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.

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.

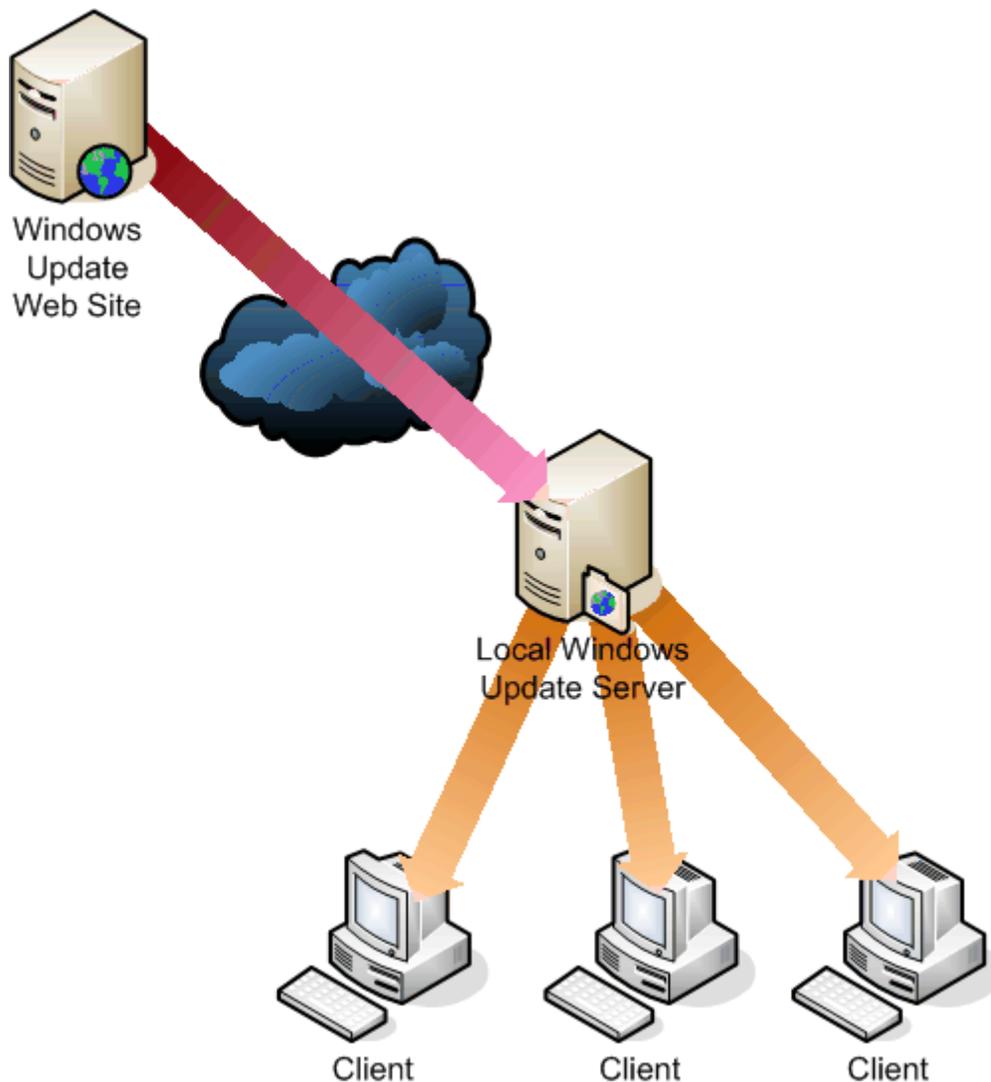


Figure 1.6: Using Microsoft's free patch management software to deploy Windows updates.

 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.

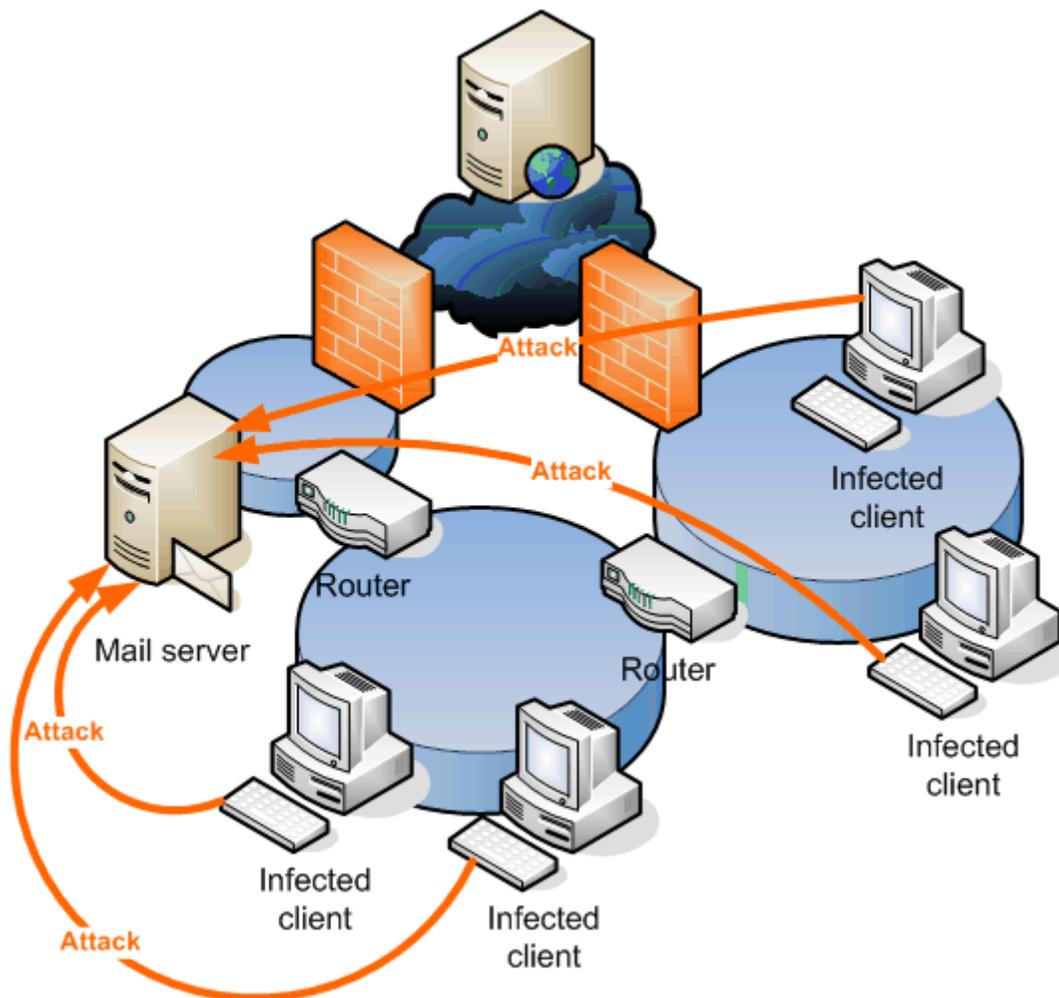


Figure 1.8: *Infected clients become launching points for a coordinated internal attack*

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.

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.

```
strComputer = "Server2"

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
```

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.

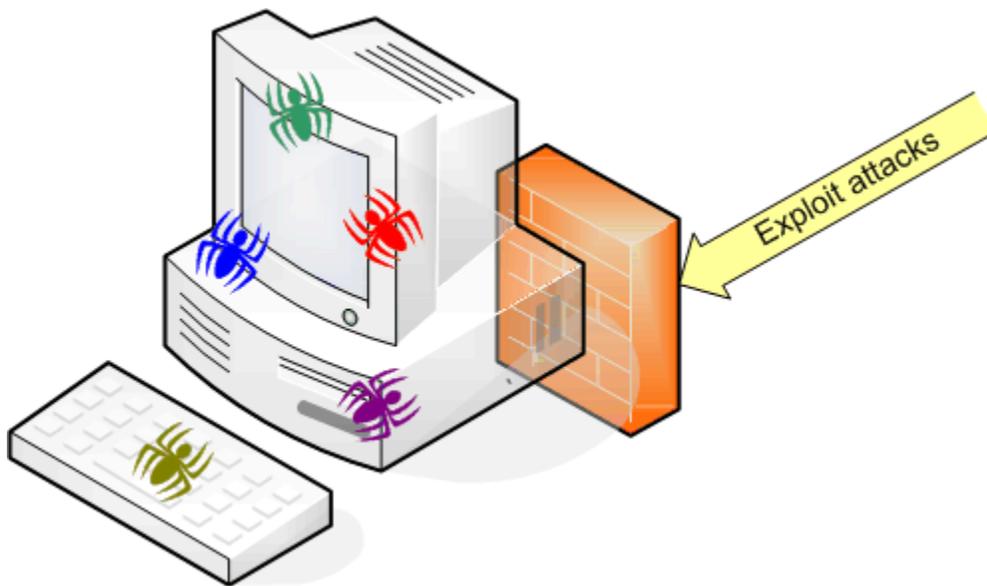


Figure 1.9: Local firewalls can help stop exploitation of built-in vulnerabilities.

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.

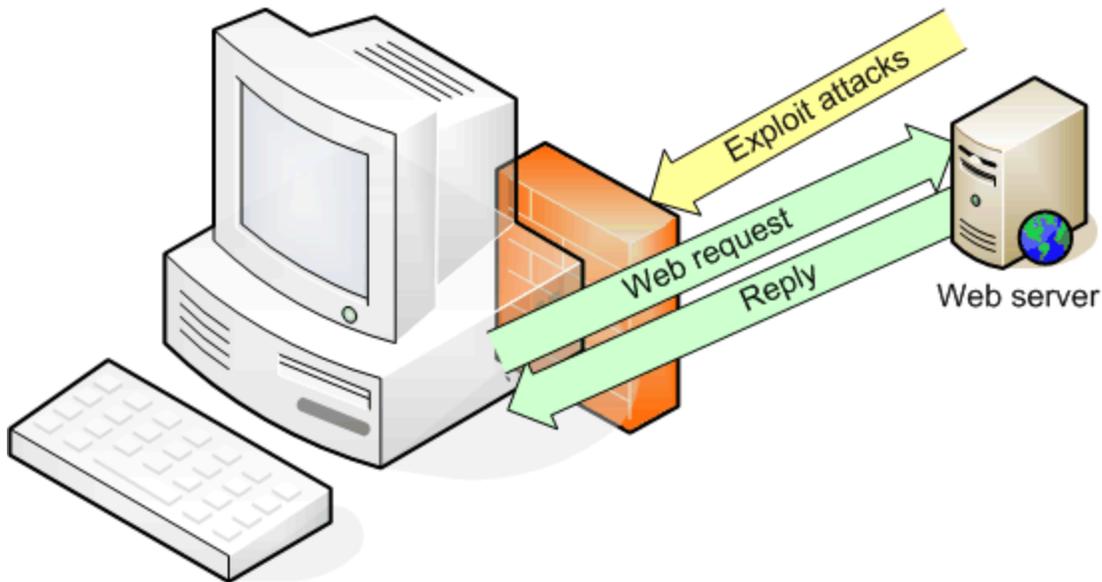


Figure 1.10: Stateful inspection allows replies in automatically.

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.

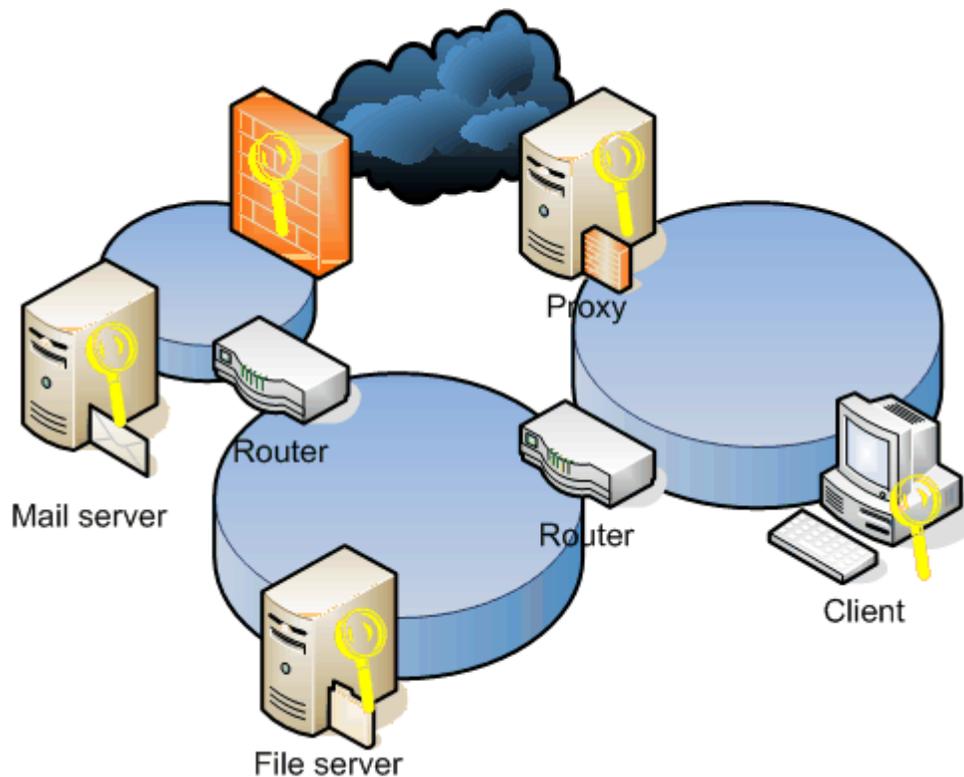


Figure 1.11: Install virus scanners everywhere on your network.

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.

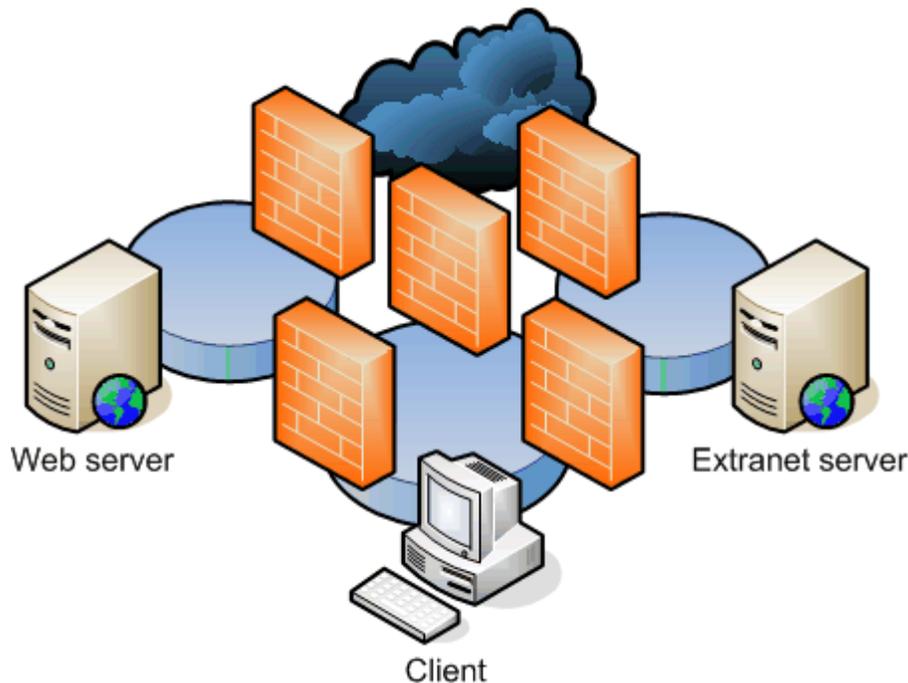


Figure 1.12: Multiple firewalls provide better protection than one.

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.

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, 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 eBook Chapters!

If you found this eBook chapter to be informative, please visit Content Central and download other eBook chapters from this publication. If you are not already a registered user of Content Central, please take a moment to register in order to gain free access to this and many other great IT eBooks and video guides. Please visit:

<http://www.realtimepublishers.com/contentcentral/>.