realtimepublishers.com®

The Definitive Guide™ To

Securing Windows in the Enterprise

SCRIPTLOGIC

Don Jones

Copyright Statement

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.

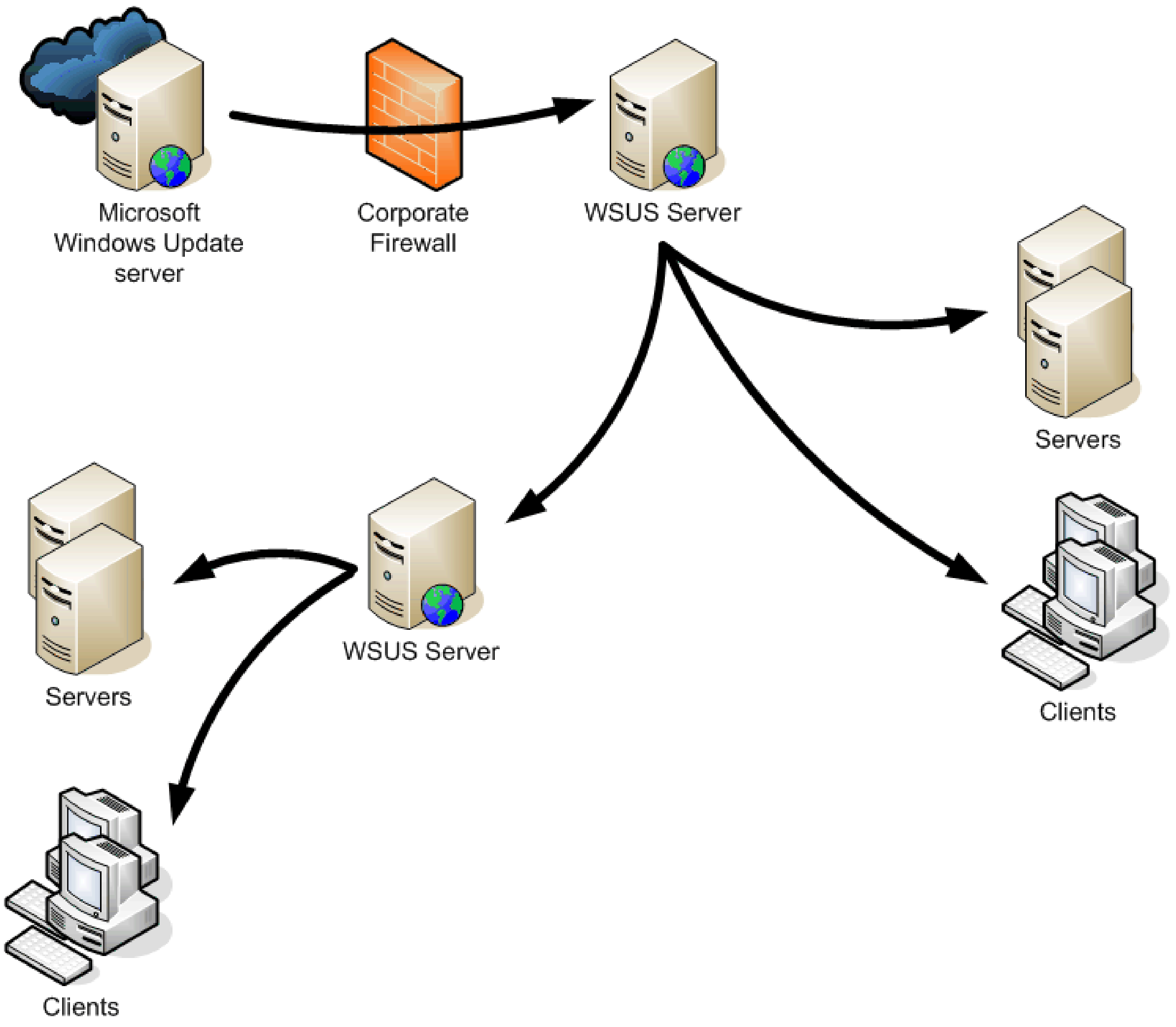


Figure 7.1: WSUS deployment.

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

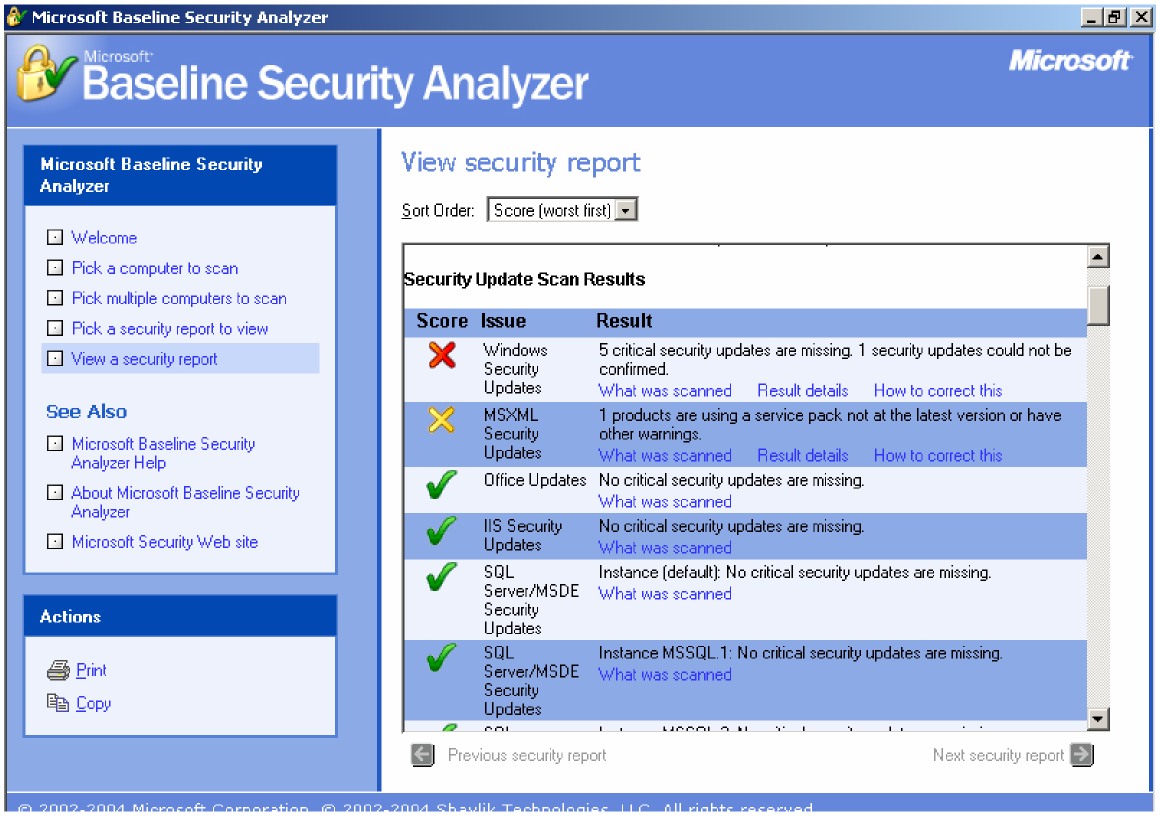
Another free tool from Microsoft, the Microsoft Baseline Security Analyzer (MBSA) is designed to scan the local computer, or one or more remote computers, and report on a number of security areas. One major are is patches; other areas include security-related configurations, and so forth.

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.


Microsoft Baseline Security Analyzer
Microsoft Baseline Security Analyzer
Microsoft
Microsoft Baseline Security Analyzer
Welcome
Pick a computer to scan
Pick multiple computers to scan
Pick a security report to view
View a security report
See Also
Microsoft Baseline Security Analyzer Help
About Microsoft Baseline Security Analyzer
Microsoft Security Web site
Actions
Print
Copy
Previous security report
Next security report

View security report

Sort Order: Score (worst first)

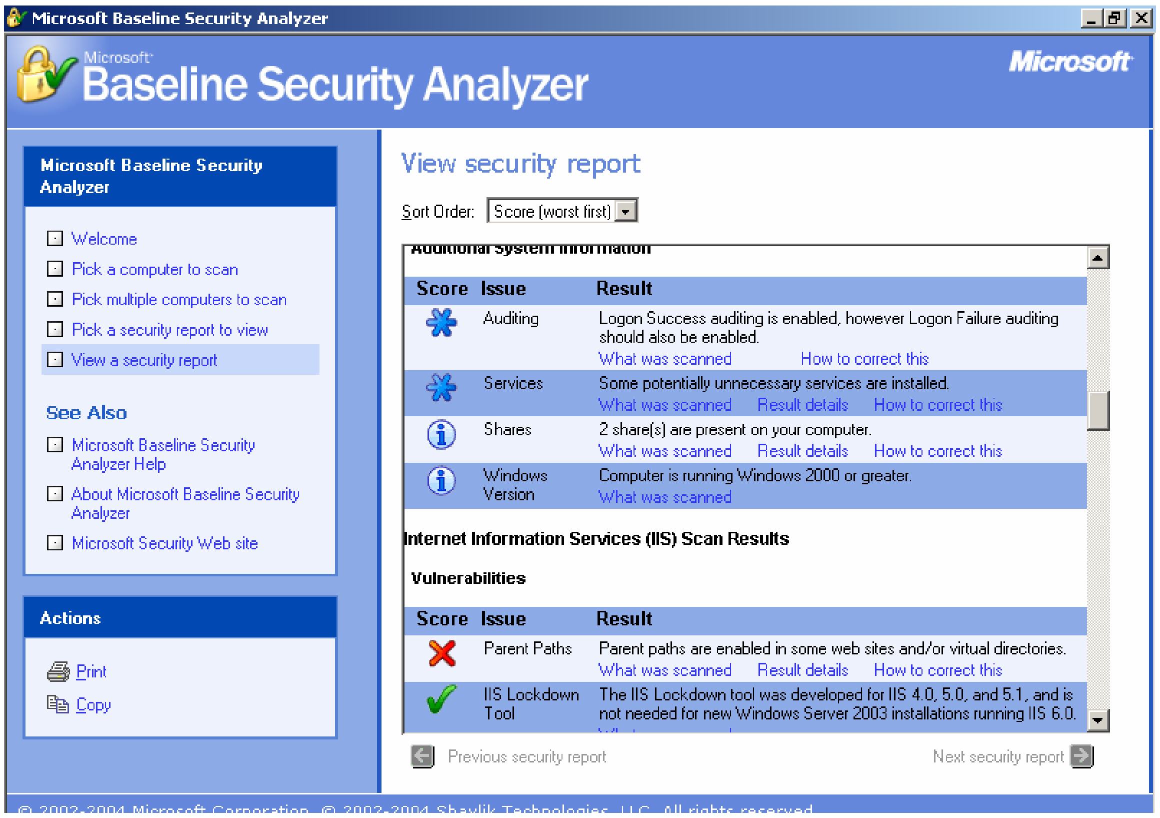
Security Update Scan Results

Score	Issue	Result
✗	Windows Security Updates	5 critical security updates are missing. 1 security updates could not be confirmed. What was scanned Result details How to correct this
✗	MSXML Security Updates	1 products are using a service pack not at the latest version or have other warnings. What was scanned Result details How to correct this
✓	Office Updates	No critical security updates are missing. What was scanned
✓	IIS Security Updates	No critical security updates are missing. What was scanned
✓	SQL Server/MSDE Security Updates	Instance (default): No critical security updates are missing. What was scanned
✓	SQL Server/MSDE Security Updates	Instance MSSQL.1: No critical security updates are missing. What was scanned

© 2002-2004 Microsoft Corporation. © 2002-2004 Shavlik Technologies, LLC. All rights reserved.

Figure 7.2: Viewing MBSA results for security patches.

However, MBSA isn't limited to patch management (see Figure 7.3).


Microsoft Baseline Security Analyzer
Microsoft Baseline Security Analyzer
Microsoft
Microsoft Baseline Security Analyzer
Welcome
Pick a computer to scan
Pick multiple computers to scan
Pick a security report to view
View a security report
See Also
Microsoft Baseline Security Analyzer Help
About Microsoft Baseline Security Analyzer
Microsoft Security Web site
Actions
Print
Copy
View security report
Previous security report
Next security report
© 2002-2004 Microsoft Corporation. © 2002-2004 Shavlik Technologies, LLC. All rights reserved.

Sort Order: Score (worst first)

Additional System Information

Score	Issue	Result
✱	Auditing	Logon Success auditing is enabled, however Logon Failure auditing should also be enabled. What was scanned How to correct this
✱	Services	Some potentially unnecessary services are installed. What was scanned Result details How to correct this
ℹ	Shares	2 share(s) are present on your computer. What was scanned Result details How to correct this
ℹ	Windows Version	Computer is running Windows 2000 or greater. What was scanned

Internet Information Services (IIS) Scan Results

Vulnerabilities

Score	Issue	Result
✗	Parent Paths	Parent paths are enabled in some web sites and/or virtual directories. What was scanned Result details How to correct this
✓	IIS Lockdown Tool	The IIS Lockdown tool was developed for IIS 4.0, 5.0, and 5.1, and is not needed for new Windows Server 2003 installations running IIS 6.0.

Figure 7.3: MBSA also analyzes security configurations.

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.

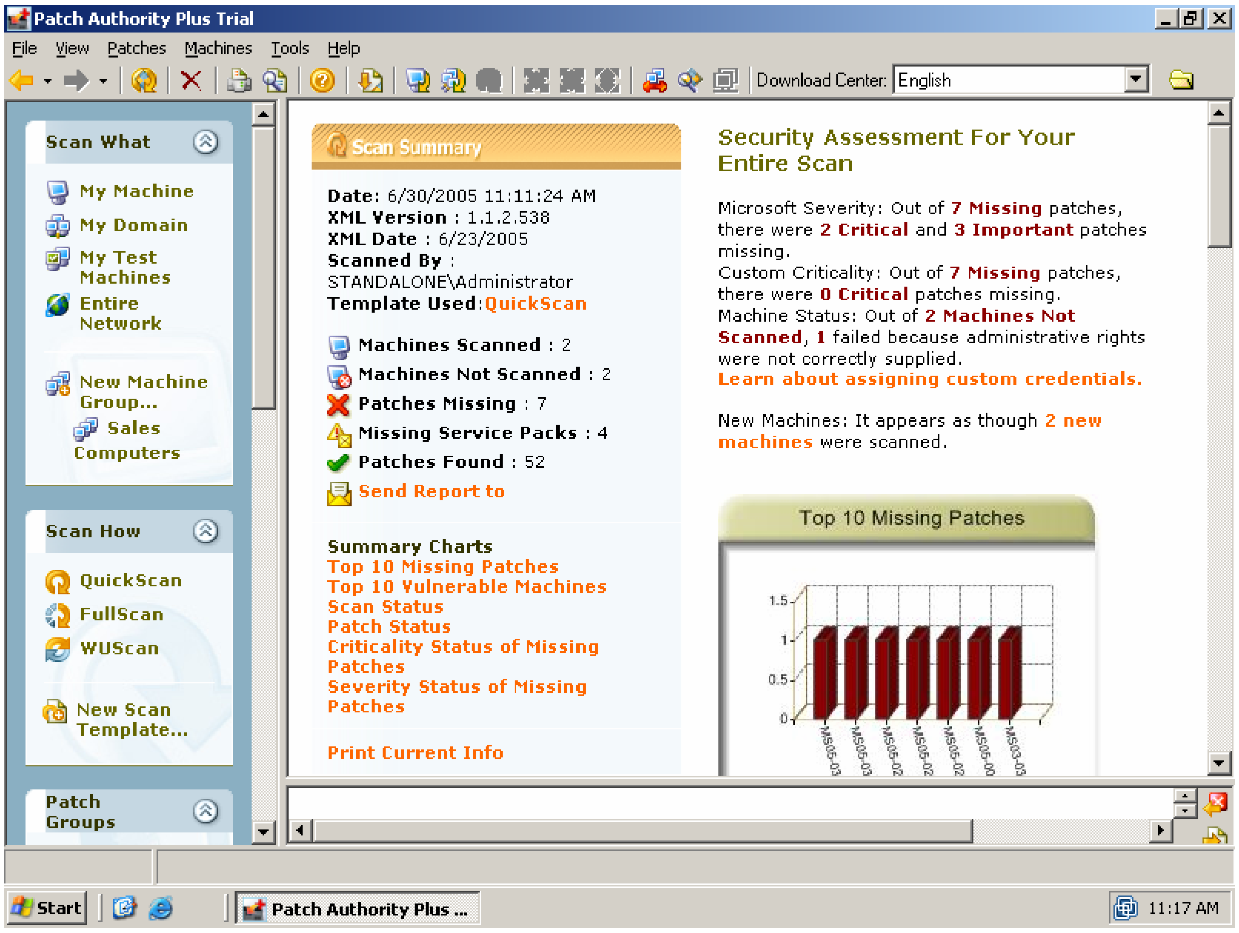
In an agentless architecture, no software resides on managed machines. This setup is similar to MBSA. A downside is that more network bandwidth is typically consumed because the entire scan is done from a remote computer over the network; however, there is no additional software to deploy and manage.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



Date: 6/30/2005 11:11:24 AM
XML Version : 1.1.2.538
XML Date : 6/23/2005
Scanned By : STANDALONE\Administrator
Template Used: QuickScan

Machines Scanned : 2
Machines Not Scanned : 2
Patches Missing : 7
Missing Service Packs : 4
Patches Found : 52
Send Report to

Summary Charts
Top 10 Missing Patches
Top 10 Vulnerable Machines
Scan Status
Patch Status
Criticality Status of Missing Patches
Severity Status of Missing Patches

Print Current Info

Security Assessment For Your Entire Scan

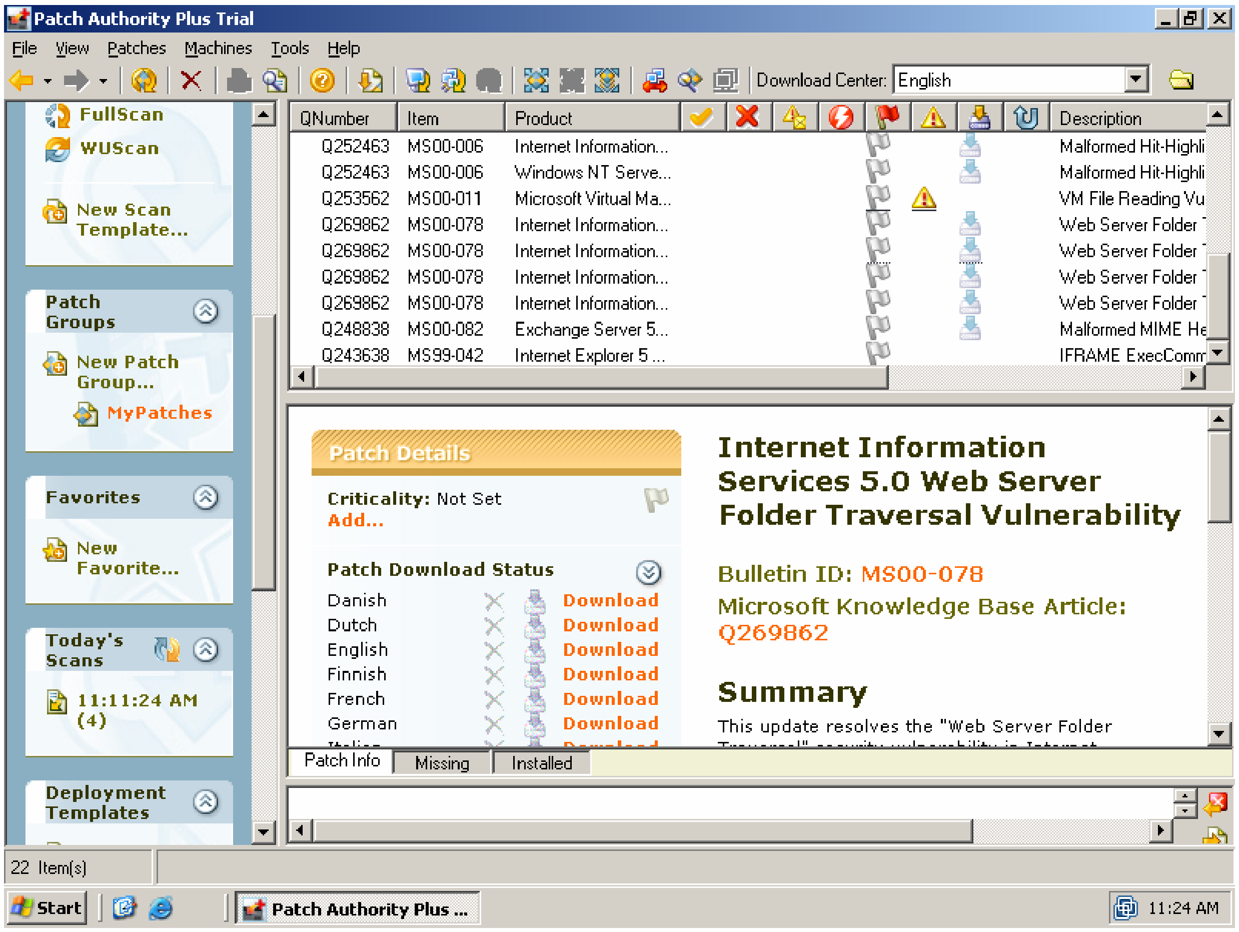
Microsoft Severity: Out of **7 Missing** patches, there were **2 Critical** and **3 Important** patches missing.
Custom Criticality: Out of **7 Missing** patches, there were **0 Critical** patches missing.
Machine Status: Out of **2 Machines Not Scanned**, **1** failed because administrative rights were not correctly supplied.
Learn about assigning custom credentials.

New Machines: It appears as though **2 new machines** were scanned.

Figure 7.4: Patch Authority Plus' scan summary.

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.



QNumber	Item	Product	Description
Q252463	MS00-006	Internet Information...	Malformed Hit-Highli
Q252463	MS00-006	Windows NT Serve...	Malformed Hit-Highli
Q253562	MS00-011	Microsoft Virtual Ma...	VM File Reading Vu
Q269862	MS00-078	Internet Information...	Web Server Folder
Q269862	MS00-078	Internet Information...	Web Server Folder
Q269862	MS00-078	Internet Information...	Web Server Folder
Q269862	MS00-078	Internet Information...	Web Server Folder
Q248838	MS00-082	Exchange Server 5...	Malformed MIME He
Q243638	MS99-042	Internet Explorer 5 ...	IFRAME ExecComm

Patch Details

Criticality: Not Set
Add...

Patch Download Status

Danish	Download
Dutch	Download
English	Download
Finnish	Download
French	Download
German	Download

Internet Information Services 5.0 Web Server Folder Traversal Vulnerability

Bulletin ID: MS00-078

Microsoft Knowledge Base Article: Q269862

Summary

This update resolves the "Web Server Folder

Figure 7.5: Patch details in Patch Authority Plus.

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).

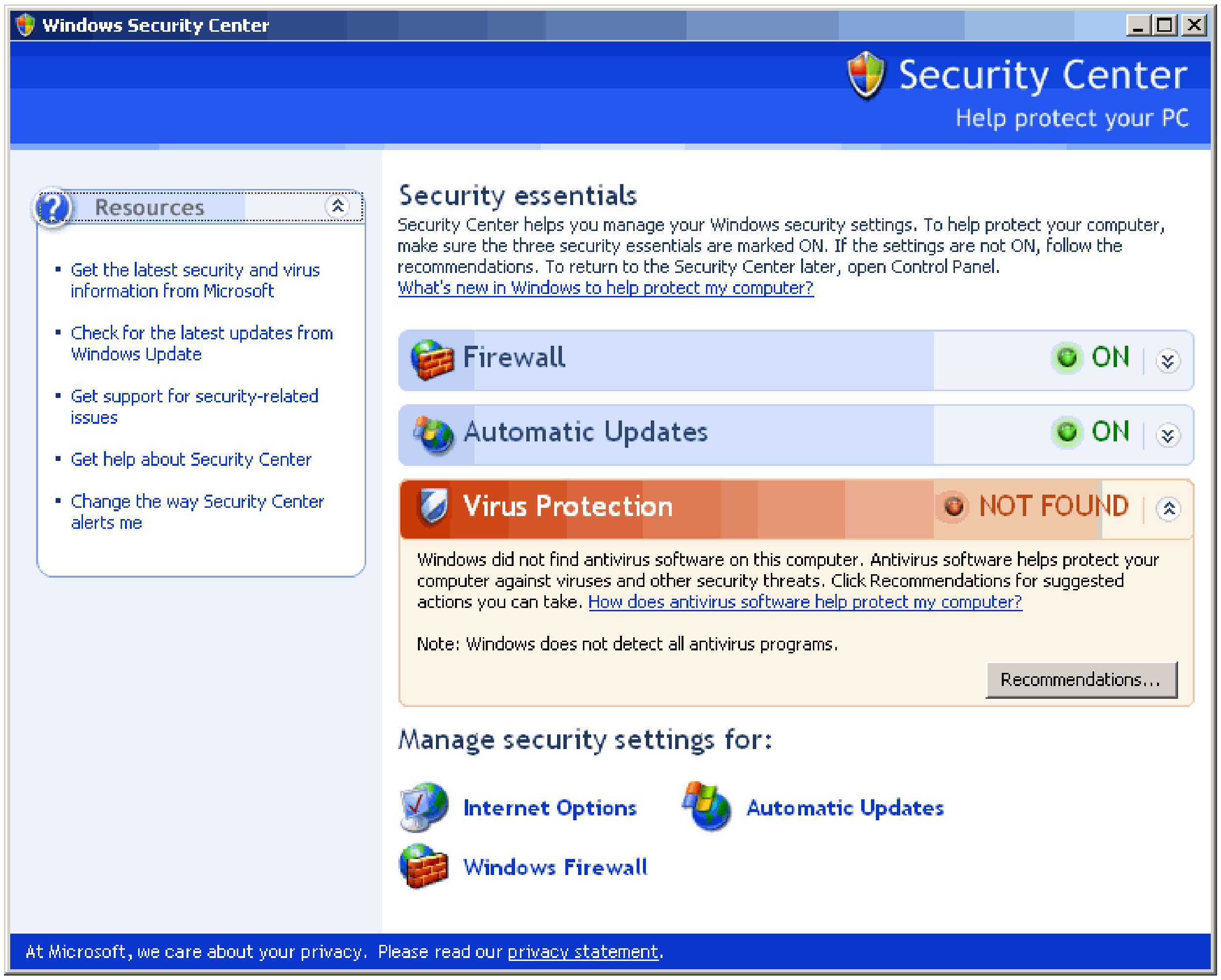
Security Software Maintenance

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.



Windows Security Center

Security Center
Help protect your PC

Resources

- Get the latest security and virus information from Microsoft
- Check for the latest updates from Windows Update
- Get support for security-related issues
- Get help about Security Center
- Change the way Security Center alerts me

Security essentials

Security Center helps you manage your Windows security settings. To help protect your computer, make sure the three security essentials are marked ON. If the settings are not ON, follow the recommendations. To return to the Security Center later, open Control Panel.
What's new in Windows to help protect my computer?

Firewall ON

Automatic Updates ON

Virus Protection NOT FOUND

Windows did not find antivirus software on this computer. Antivirus software helps protect your computer against viruses and other security threats. Click Recommendations for suggested actions you can take. How does antivirus software help protect my computer?

Note: Windows does not detect all antivirus programs.

Recommendations...

Manage security settings for:

Internet Options
Automatic Updates
Windows Firewall

At Microsoft, we care about your privacy. Please read our privacy statement.

Figure 7.6: Windows Security Center.

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.

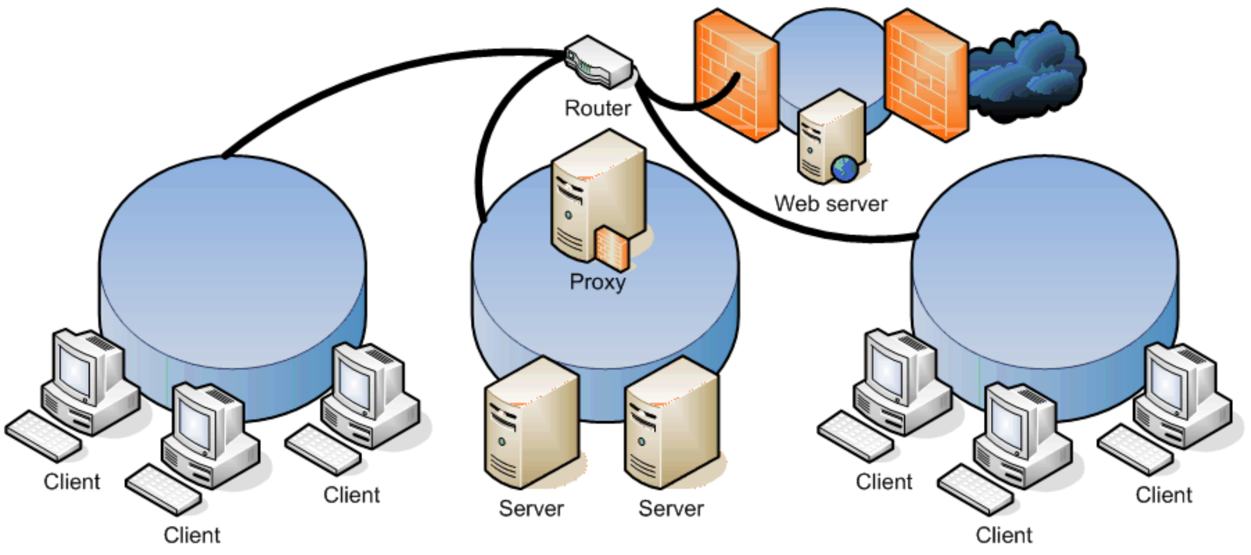


Figure 7.7: A typical corporate network.

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.

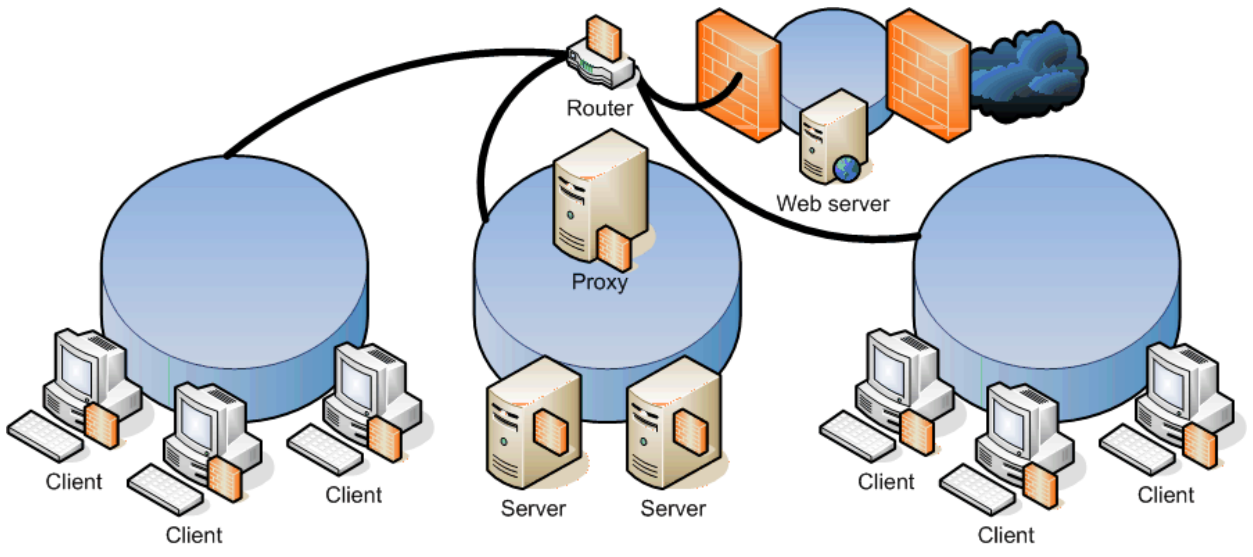


Figure 7.8: Adding a multi-layered defense.

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.

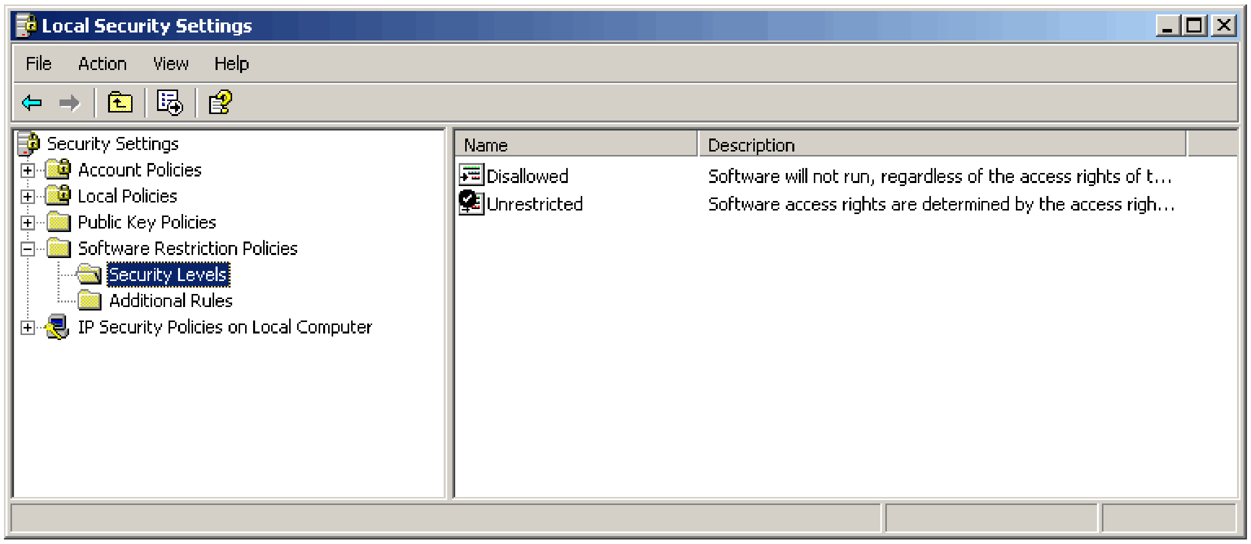
What is Software Filtering?

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.



Name	Description
Disallowed	Software will not run, regardless of the access rights of t...
Unrestricted	Software access rights are determined by the access righ...

Figure 7.9: Default SRPs.

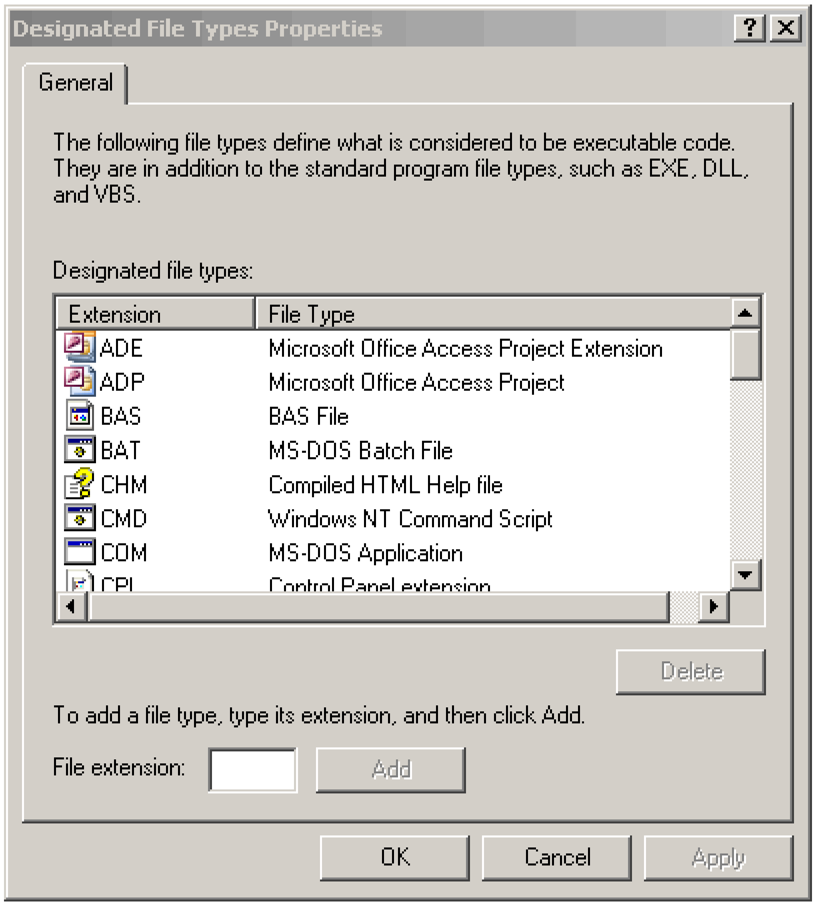
However, as I mentioned earlier, trying to identify all the *bad* software in the world would be practically impossible. Better to switch the system to use Disallowed as its default, then simply identify the software you *want* running on your computers.

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.



The following file types define what is considered to be executable code. They are in addition to the standard program file types, such as EXE, DLL, and VBS.

Designated file types:

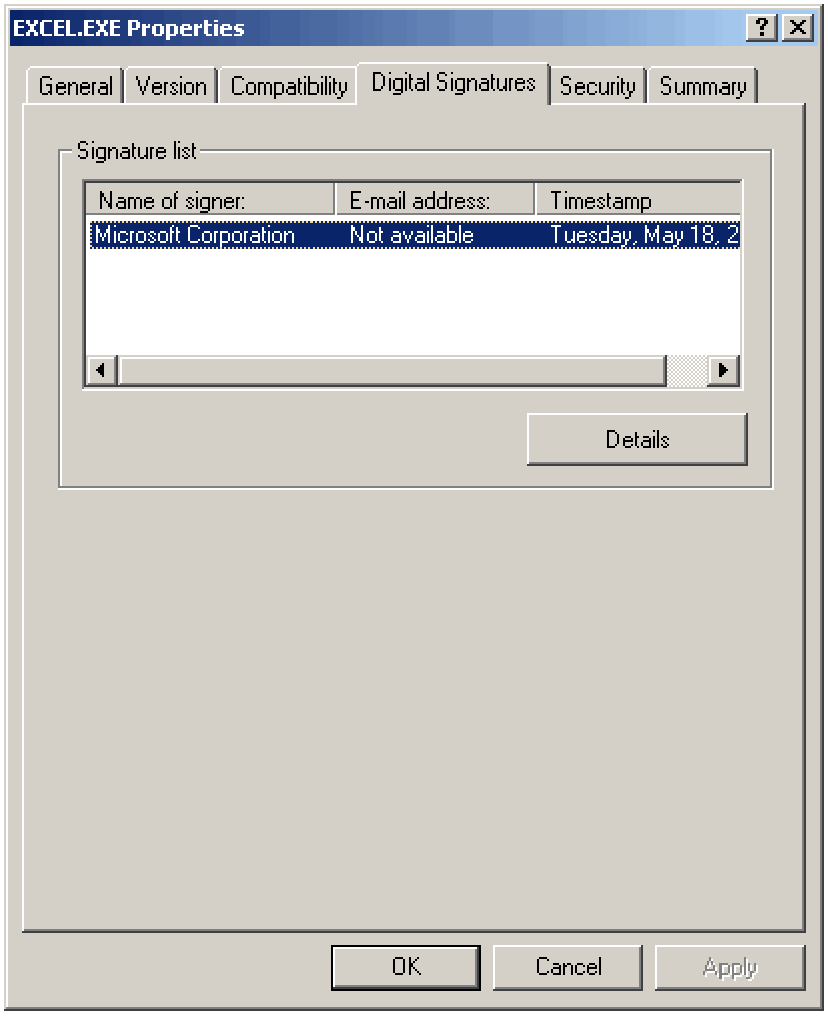
Extension	File Type
ADE	Microsoft Office Access Project Extension
ADP	Microsoft Office Access Project
BAS	BAS File
BAT	MS-DOS Batch File
CHM	Compiled HTML Help file
CMD	Windows NT Command Script
COM	MS-DOS Application
CPL	Control Panel extension

To add a file type, type its extension, and then click Add.

Figure 7.10: SRP's default list of executables.

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.

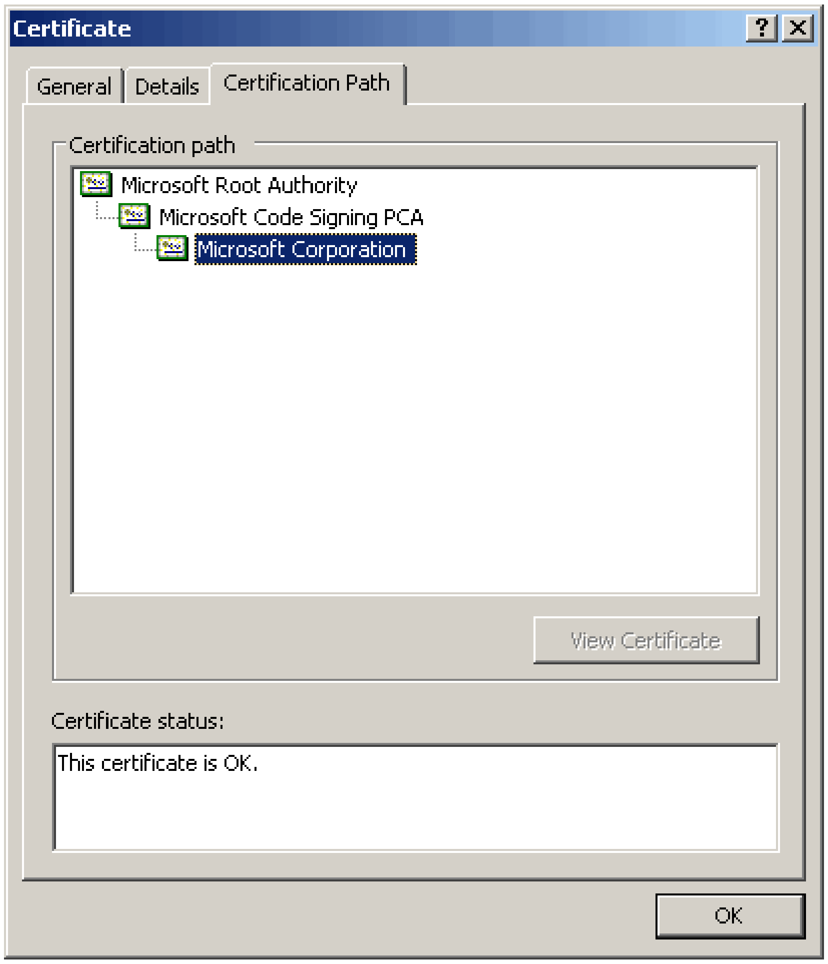
Software can be digitally signed. Examining the properties of Excel.exe, for example, you can see that it has been digitally signed by Microsoft.com (see Figure 7.11).



Name of signer:	E-mail address:	Timestamp
Microsoft Corporation	Not available	Tuesday, May 18, 2

Figure 7.11: Digital signature on Excel.exe.

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.

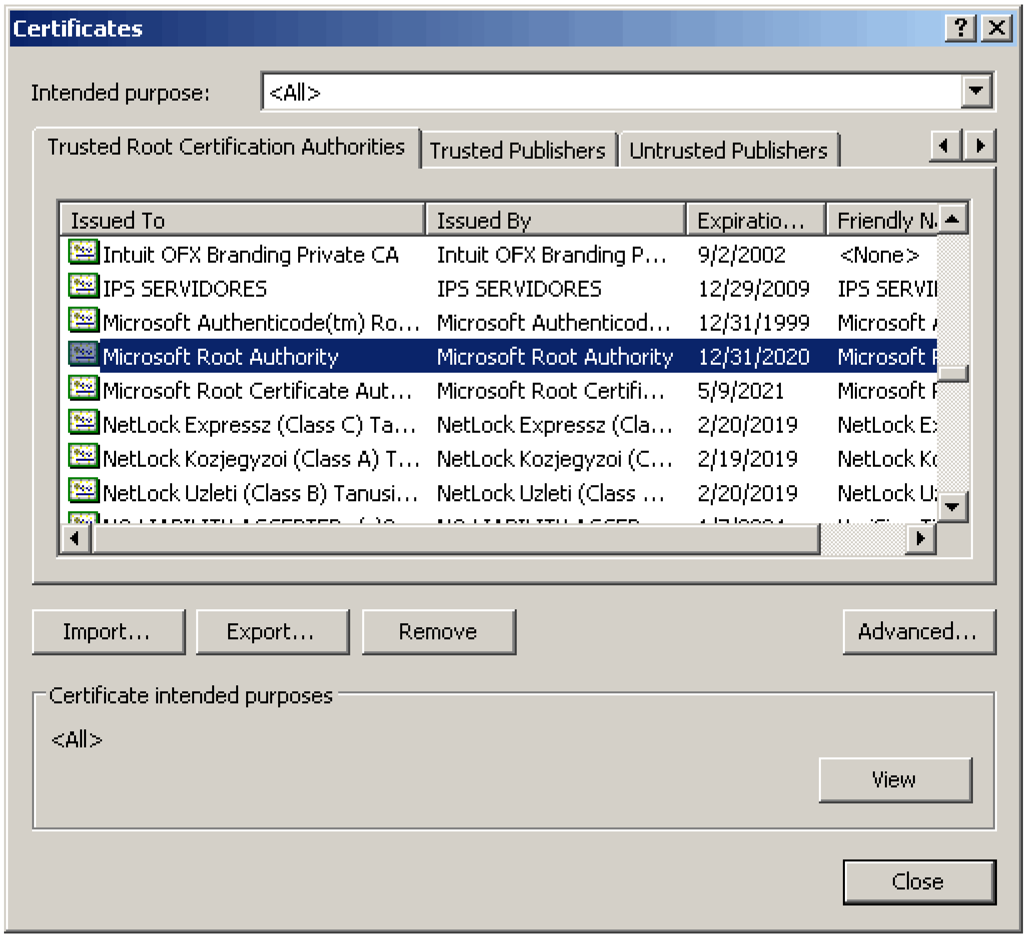


Microsoft Root Authority
Microsoft Code Signing PCA
Microsoft Corporation

This certificate is OK.

Figure 7.12: Determining the publisher.

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.



Issued To	Issued By	Expiratio...	Friendly N
Intuit OFX Branding Private CA	Intuit OFX Branding P...	9/2/2002	<None>
IPS SERVIDORES	IPS SERVIDORES	12/29/2009	IPS SERVI
Microsoft Authenticode(tm) Ro...	Microsoft Authenticod...	12/31/1999	Microsoft
Microsoft Root Authority	Microsoft Root Authority	12/31/2020	Microsoft
Microsoft Root Certificate Aut...	Microsoft Root Certifi...	5/9/2021	Microsoft
NetLock Expressz (Class C) Ta...	NetLock Expressz (Cla...	2/20/2019	NetLock E
NetLock Kozjegyzoi (Class A) T...	NetLock Kozjegyzoi (C...	2/19/2019	NetLock K
NetLock Uzleti (Class B) Tanusi...	NetLock Uzleti (Class ...	2/20/2019	NetLock U

Certificate intended purposes

<All>

Figure 7.13: Viewing trusted publishers.

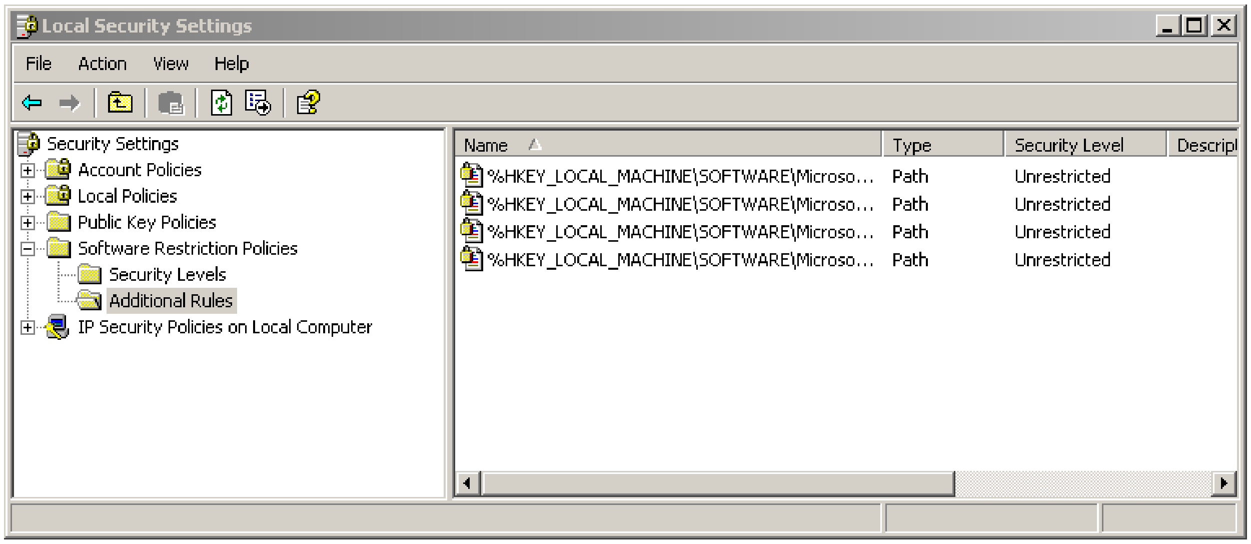
So what does that mean? It means any software that has been signed using a certificate issued by one of these trusted publishers is a trusted piece of software. This concept is central to the way SRP works. Microsoft pre-installs more than *one hundred* root certificates with Windows.

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.

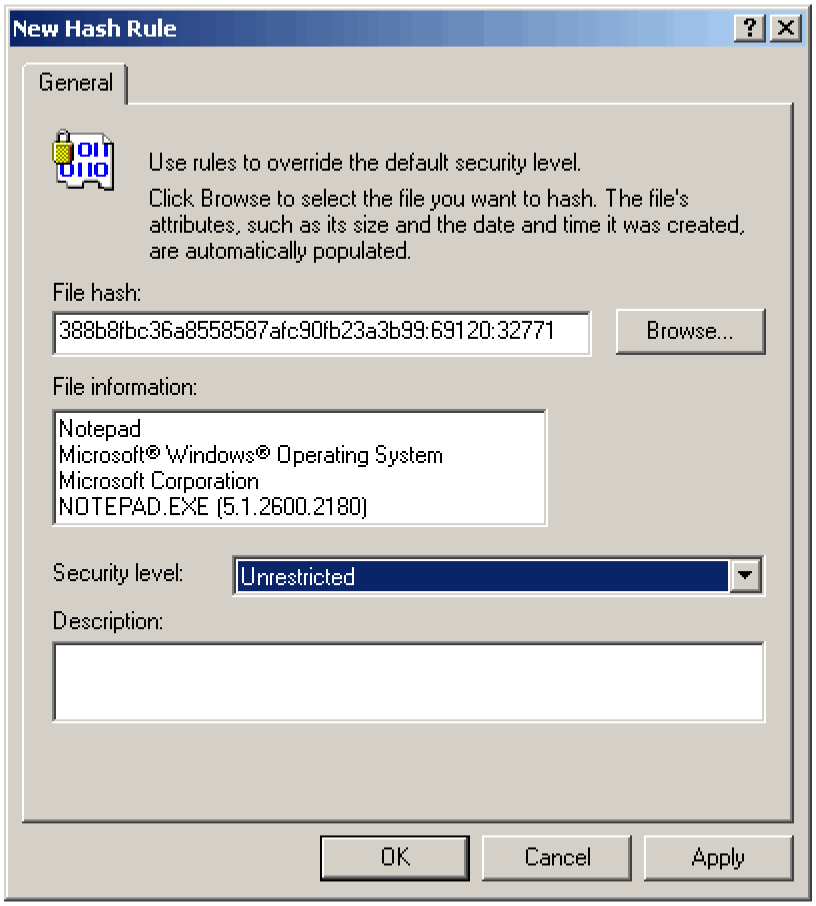
Okay, so how does all this apply to SRP? Through its *additional rules*. These are the rules that identify software that should be exempted from the default security level. SRP comes with four *path rules*, which exempt all software in the paths used by Windows (such as C:\Windows and C:\Windows\System32). 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.



Name	Type	Security Level	Descripl
%HKEY_LOCAL_MACHINE\SOFTWARE\Microso...	Path	Unrestricted	
%HKEY_LOCAL_MACHINE\SOFTWARE\Microso...	Path	Unrestricted	
%HKEY_LOCAL_MACHINE\SOFTWARE\Microso...	Path	Unrestricted	
%HKEY_LOCAL_MACHINE\SOFTWARE\Microso...	Path	Unrestricted	

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.


New Hash Rule

General

Use rules to override the default security level.

Click Browse to select the file you want to hash. The file's attributes, such as its size and the date and time it was created, are automatically populated.

File hash: 388b8fbc36a8558587afc90fb23a3b99:69120:32771

File information:
Notepad
Microsoft® Windows® Operating System
Microsoft Corporation
NOTEPAD.EXE (5.1.2600.2180)

Security level: Unrestricted

Description:

Figure 7.15: A hash rule for Notepad.exe.

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.

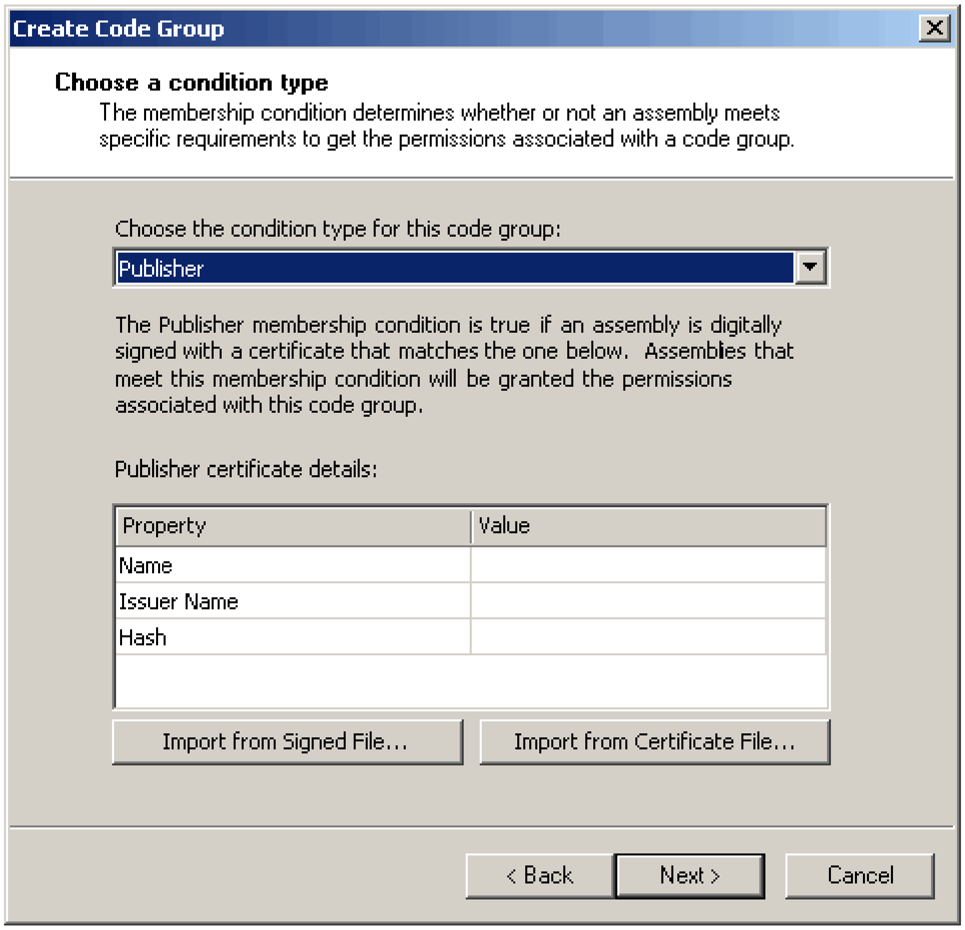
This technique has a side benefit: carefully control access to the certificate and you can prevent new, unauthorized versions of the software from running, too. It's an effective way to help control change in your environment.

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

Choose a condition type

The membership condition determines whether or not an assembly meets specific requirements to get the permissions associated with a code group.

Choose the condition type for this code group:

Publisher

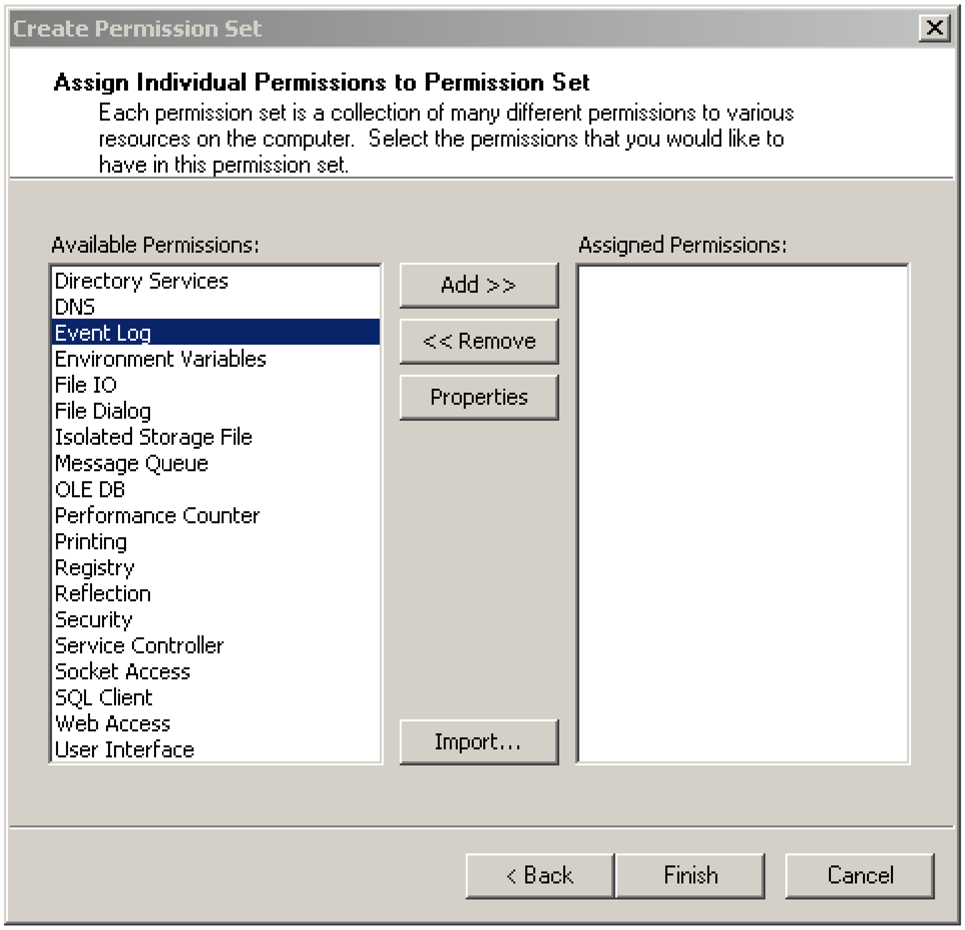
The Publisher membership condition is true if an assembly is digitally signed with a certificate that matches the one below. Assemblies that meet this membership condition will be granted the permissions associated with this code group.

Publisher certificate details:

Property	Value
Name	
Issuer Name	
Hash	

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.



Assign Individual Permissions to Permission Set

Each permission set is a collection of many different permissions to various resources on the computer. Select the permissions that you would like to have in this permission set.

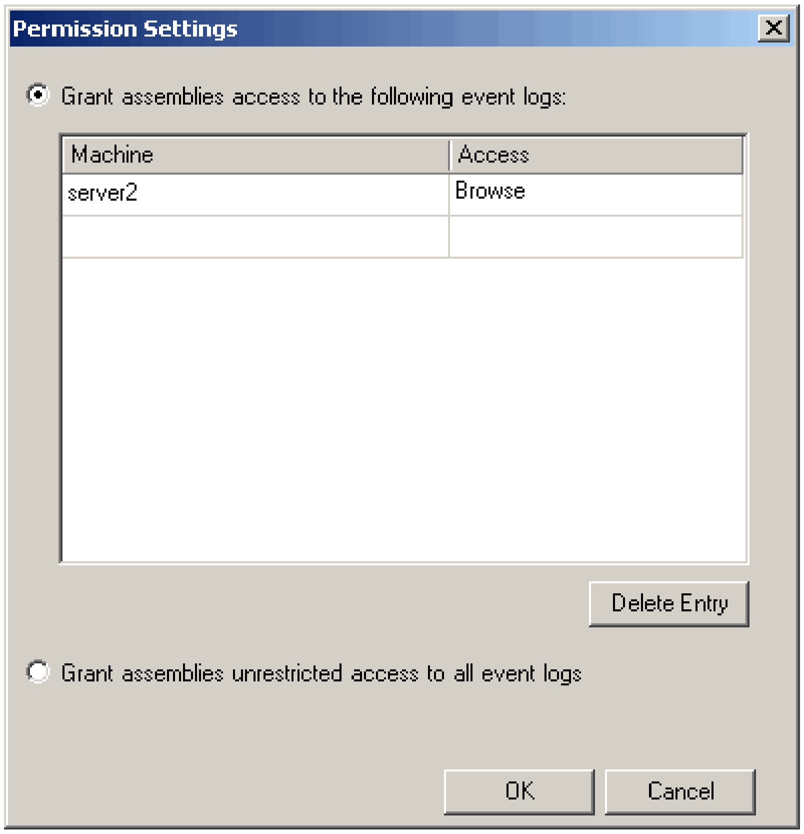
Available Permissions:

Directory Services
DNS
Event Log
Environment Variables
File IO
File Dialog
Isolated Storage File
Message Queue
OLE DB
Performance Counter
Printing
Registry
Reflection
Security
Service Controller
Socket Access
SQL Client
Web Access
User Interface

Assigned Permissions:

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.


Permission Settings

(●) Grant assemblies access to the following event logs:

Machine	Access
server2	Browse

Figure 7.18: Details of a permission set entry.

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 that 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.

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.