

ITProTM
SERIES

WindowsITPro

 **eBooks**

Keeping Your Business
SAFE from Attack:

Patch Management

By Jeff Felling

Microsoft[®]



Contents

Chapter 1 Introduction to Patch Management	1
Building the Foundation: Processes, Software, and Training	2
Processes	2
Create a Patch Management Triage and Deployment Team	2
Determine SLAs for Different Levels of Patches	5
Ensure that the Appropriate Groups Test and Sign Off on a Patch	5
Subscribe to Patch and Security Advisories and Bulletins	6
Review All New Security Bulletins with the Team to	
Assess Risk and Triage Deployment	8
Weigh Deploying Updates vs. Exploit Mitigation Efforts	9
Choosing Software to Deploy Patches	9
Windows Automatic Updates	10
Microsoft Software Update Services and Windows Update Services	11
Microsoft SMS 2003	12
Beyond Microsoft	13
Training	14
The Full Rally	15

Chapter 1:

Introduction to Patch Management

Due to the rapid proliferation of nefarious worms, with names such as MS Blaster, Nimda, and Code Red, applying Microsoft Security Updates is becoming a staple of any business connected to the Internet or outside world. However, hackers and crackers will continue to exploit computer software and your company will always need information security protection from zero-day exploits. However, a majority of the fast-spreading, heavy-hitting worms leveraged and exploited weaknesses in software that were previously identified and fixed weeks—in some cases months—earlier. Target damage aside, the proliferation of these worms affects the Internet by clogging routers and Internet gateways. In all, these worms have sent a loud-and-clear wakeup call to IT departments everywhere to get serious about patch management.

To reduce the shellshock of frequent patch releases, Microsoft continues to introduce software and processes to help triage and deploy their Security Updates. Microsoft formalized the Security Updates release cycle to occur on the second Tuesday of every month. All Security Updates are ranked in severity and classified by products. They also include detailed descriptions of the exploit and list mitigating factors. Microsoft also released several patch deployment software products in addition to the flood of new third-party patch management software products. These software products exist to help test and deploy all the patches. Most patch management software supports Microsoft products and some extends to third-party software as well.

However, the process of deploying the patches is only the tip of the iceberg. A successful and comprehensive patch management program combines well-defined processes, effective software, and training into a strategic program for assessing, triaging, obtaining, testing, and deploying software patches. Patching software is not a new phenomenon: software updates are a frequent and regular occurrence and historically patches improved performance, stability, or even added new program features. But of late, the proliferation of Internet worms and viruses have put the spotlight on patch management vis-à-vis Microsoft Security Updates. The rapid assessment and successful deployment of these Security Updates causes the most anxiety in IT shops throughout the world. These shops must balance the potential threats to unpatched systems, project priority, time necessary to identify and assess security vulnerabilities, and the testing and deployment of patches with the potential business impact of patch installation (e.g., reboot downtime, unsuccessful patch deployment).

This book describes attributes of a successful patch management program and explains Microsoft's update technologies and security update communications network. Your internal processes coupled with Microsoft's evolving update distribution program will define your patch management program. Partially due to the recent attention drawn to the Security Updates, Microsoft continues to improve its security update communications. The latest bulletins describe the updates in sufficient detail to help most organizations identify and triage patches relevant to their environment.

This text will also outline how to assemble a patch testing program that calls on the expertise of resources across your enterprise to minimize adverse effects that a patch might have on your network's business-critical systems and applications. You'll learn how to set up a patch testing program

2 Keeping Your Business Safe from Attack: Patch Management

that provides an important safety net for your production servers. The later chapters will examine the Microsoft patch mechanisms and Microsoft's update distribution software: Windows Update, Windows Update Server, and Systems Management Server (SMS) 2003.

Building the Foundation: Processes, Software, and Training

Let's look at what constitutes a solid patch management program. The details vary by organization but traits common to all successful programs include:

- Identifying the processes to assess, test, deploy, and audit the patch installation
- Selecting effective patch testing and distribution software for your organization, then using this software to deploy the updates
- Training to ensure that everyone is capable and ready to test and deploy patches when the time comes
- Gaining support from executive management that includes sponsorship and setting overall goals for patch management

Processes

The patch management process defines the strategy and tactics encompassing your patching program and includes activities ranging from the selection and deployment of patch management software, to creating a Patch Management Triage and Deployment Team, to rolling out the individual patches.

Customize each of these elements for your particular organizational needs. Smaller organizations might not have a formal process but will benefit from a structured approach nonetheless. Be sure to include in your process early planning topics such as researching, purchasing, and deploying the patch delivery software for each of your organization's locations, including branch offices and remote users. Consider these elements when defining your patch management processes:

- Create a Patch Management Triage and Deployment Team.
- Subscribe to Microsoft and non-Microsoft patch and security advisories and bulletins.
- Review all new security bulletins with the team to assess risk and triage deployment of new patches or evaluate workarounds.
- Weigh deploying updates versus exploit mitigation efforts for different patches, environments, or targets.
- Determine service level agreements (SLAs) for different patch levels, such as internal versus production or workstation versus server.
- Devise and document testing procedures to ensure that the appropriate groups test and sign off on a patch before it's released to production. When feasible, consider a *burn in period* in which the patch is tested in a live yet limited environment.

Create a Patch Management Triage and Deployment Team

Effective emergency response or disaster recovery teams drill repeatedly so that when the time comes they are prepared to handle the event. This training is no different from an Information Security alert team tasked with investigating unknown events or attacks. Adopting the effective strategies of these emergency response teams is becoming more important for your patch deployment team. Critical patch deployments increasingly require fast action—especially when an exploit is in the wild.

In many organizations, the patch deployment team consists of systems administrators or engineers who have primary responsibilities beyond patching systems. Since the burst of the dot-com

bubble in 2000, most IT spending budgets have shrunk and resources have thinned considerably. In many companies, the IT staff is being asked to do more with less help, which unfortunately can mean that nonrevenue or maintenance activities might be unintentionally (or purposely) reprioritized.

To help ensure that patching is not an afterthought at your company, consider forming a Patch Management Triage and Deployment Team that includes representatives from each of the disciplines or functional areas of your organizations: Microsoft SQL Server, Microsoft Exchange Server, Active Directory, file and print, Web, custom and proprietary applications, etc. By involving subject matter experts from each of these disciplines, you make certain that when patching time comes you can rely on each expert to test and deploy the patches to their systems. Especially in large organizations, involving these folks early on helps with team building so that when a patching crisis arises response team members already know one another, which implicitly improves communication. Include Business Decision Makers (BDMs) and representative customers who can help assess system risk tolerance. The BDMs can work with the technical teams to schedule and test patches for specific business-critical systems. Customers of these systems can provide valuable insight into usage patterns for scheduling server reboots and downtime or into when workarounds would be beneficial until a patch can be applied. For large enterprises, your Patch Management Triage and Deployment Team might include multiple BDMs.

Even during times when you are not deploying patches, schedule regular weekly meetings with the team members to discuss current or upcoming patches, deployment systems, triage strategies, or general training. Schedule these reoccurring, standing meetings out into the future so that they are on key participants' calendars. Then when a patch needs a quick assessment, testing, and deployment, the right people already have the time reserved.

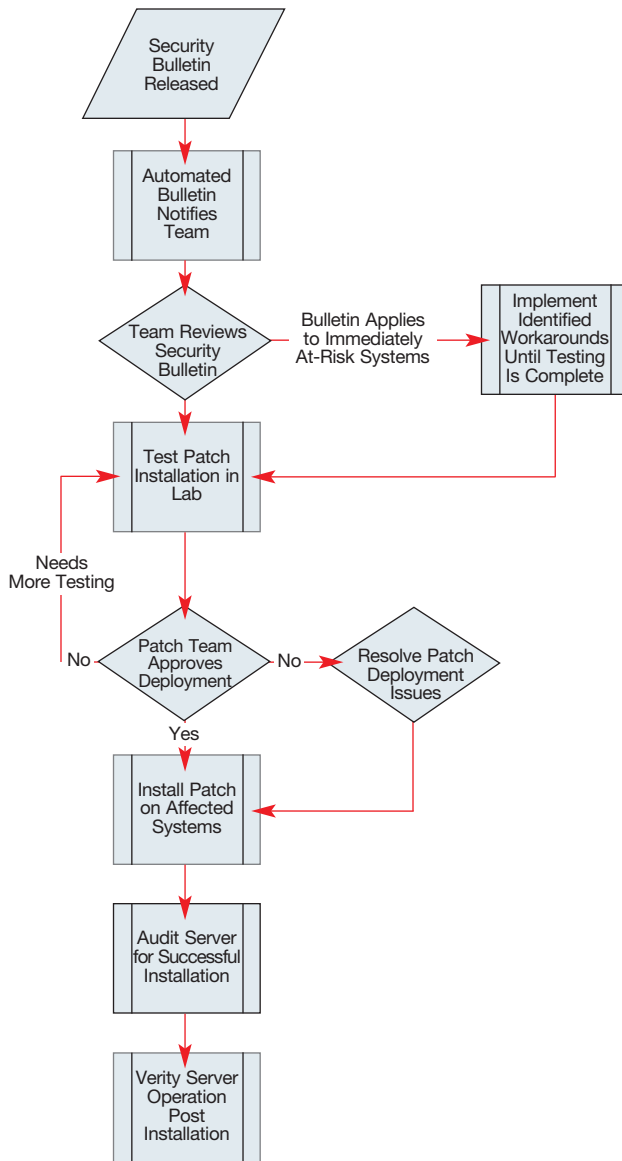
Consider establishing different states of alert for your Patch Management Triage and Deployment Team. Under normal circumstances when no patches need deployment, use the meetings to discuss or review your patch deployment technologies. Discuss upcoming projects that might tie up key patching resources, such as testing labs or deployment personnel. These meetings are also an ideal time to train your team in the process of deploying patches when necessary. Also consider developing two patch management processes, one for regular patch releases (e.g., a worm is in the wild) and one for emergency patch deployment (e.g., a worm is inside your company's network boundaries).

Of course when patches must be deployed, the primary role of the team comes into direct play. In general, the second Tuesday of every month is the day that Microsoft releases the majority of its patches for the month. Microsoft typically announces the patches by noon PST, so Tuesday afternoons are good times to meet and be ready when Microsoft releases a new batch of updates. Note that critical patches for exploits in the wild can be released outside of this timeframe at Microsoft's discretion. For this reason, subscribing to Microsoft's free Security Update notification service is a good idea. The next section describes this service in more detail.

Upon notification of new Security Updates, rally the Patch Management Triage and Deployment Team and begin your patch management process. Assess the patches and triage their applicability and exploit risk to your environment. Figure 1-1 shows a sample process.

For example, you will likely handle an Internet Explorer (IE) patch differently than a core Windows OS patch such as a Local Security Authority Subsystem (LSASS) security update. The IE patch's focus might be on deployment to employee workstation computers whereas the OS patch might need immediate rollout to any Internet connected computers and possibly others depending on the specific *exploit attack vector*.

Figure 1-1
Reviewing the patch management process



The *exploit attack vector* is the mechanism an attacker uses to compromise a vulnerable system. For example, an IE exploit attack vector might be a visit to a Web site containing malicious code. This means that a user must actively visit an infected site. Depending on your organizations IE security

policy this may or may not be a critical patch to deploy to your end users. Contrast this to the vulnerability of a primary security DLL such as LSASS. This DLL is used by many externally accessible components and depending on the vulnerability, can be exploitable from an unsolicited external connection attempt via Secure Sockets Layer (SSL), remote procedure call (RPC), or other LSASS-enabled protocol. To exploit this vulnerability, an external attacker might only need network access to a vulnerable server. If an SSL-protected Web site exposes this vulnerability, then that company's Internet connected Web site might be at risk. The exploit attack vector might be anyone on the Internet establishing an SSL connection to your Web site. Worms that spread from one vulnerable server to another frequently use this type of exploit attack vector. These malicious software programs exploit an unpatched vulnerability, infect the computer, then launch new attacks from the compromised computer. Code Red, Sasser, and MS Blaster are all examples of worms that spread by exploiting vulnerabilities that had official patches available months earlier.

The Patch Management Triage and Deployment Team must consider all these factors when determining when and how quickly patches need testing and deployment. Later this chapter explains how mitigating factors can help buy your company time to conduct adequate testing of new patches. However, even with these mitigations, patching has no substitute. The time between disclosure of a vulnerability and the availability of an automated exploit shrinks every year—from more than 300 days a couple of years ago to only 17 days for the recent Sasser exploit. Chapter 3 describes techniques and processes for testing the patches and updates.

Determine SLAs for Different Levels of Patches

Let's face it, patching disrupts normal business operations and, unless your IT department is overstaffed, you will have to make concessions to other projects to accommodate your patch deployments. To acknowledge your patching activities alongside other business projects, create a policy that specifies patching SLAs that both the businesses and technical leadership approve.

Include in these SLAs definitions of different levels and types of patches (e.g., internal versus production, workstation versus server), define their priority, and set an expectation for when specific computers will be patched after the release of a new alert. A very basic SLA might assert that all patches deemed critical by Microsoft will be deployed within 48 hours and all other patches will be deployed within 2 weeks. Of course you will want to customize this to your environment and tailor it to suite your needs. A well-defined SLA will not only help ensure that patches get deployed shortly after release but they also help clear any roadblocks in securing resources to assist with the patch deployments. Plus by defining your SLAs up front, your business management will probably be more tolerant of a delayed business project milestone due to a patch deployment exercise.

Ensure that the Appropriate Groups Test and Sign Off on a Patch

You need to devise and document testing procedures for the patches. These procedures are to ensure that the appropriate groups test and sign off on a patch before released to production. You also need to consider a burn in period when feasible.

All too often—especially in the heat of battle—patches are deployed without adequate testing. Many times, administrators assume that it will work and more-or-less hope that the computer will successfully restart. Although for the most part this is true due to Microsoft's rigorous testing, a couple of patches have had serious problems. For example, the MS04-011 patch released in 2004 caused some combinations of hardware to stop responding. Although infrequent, a patch might dramatically

6 Keeping Your Business Safe from Attack: Patch Management

change how software behaves between a patched and unpatched system. An example of this was SQL Server Service Pack 3 (SP3), which implemented additional security settings that affected customer's custom application code in some circumstances.

By involving many cross-functional groups in your Patch Management Deployment and Triage Team you will have the right people on hand to perform this testing. They will be the experts who deploy the patches to their systems, then test or watch the system over a period of time to look for any anomalous behavior.

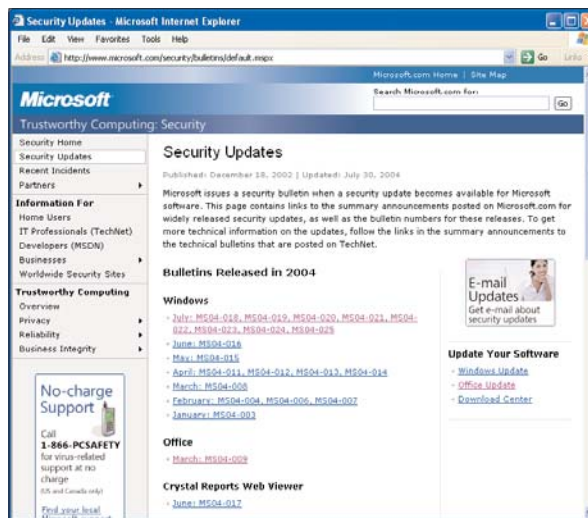
You might be able to gain flexibility for deploying your patches if you can deploy patches in stages to certain groups of servers. For example if you manage a Web farm of multiple Web servers, even after testing in a lab, consider deploying the patch to one Web server and watching it for a few days. This burn in period tests the patch in a live environment, and if no apparent problems appear, then after some time you can deploy the patch to the remaining servers with more confidence. However with a progressive type of rollout, waiting a few days can be the difference between deploying before a worm and being infected by a worm.

Chapter 3 delves into the detail aspects of testing that help create a solid testing program. Make sure to include testing in your process and training.

Subscribe to Patch and Security Advisories and Bulletins

The proliferation of worms that exploit known software vulnerabilities has spawned several patch and security advisory Web sites and bulletins. The primary Security Updates Web site for Windows is the Microsoft Security Bulletin Web site at <http://www.microsoft.com/security/bulletins>, which Figure 1-2 shows.

Figure 1-2
Viewing Microsoft's searchable Security Updates Web site



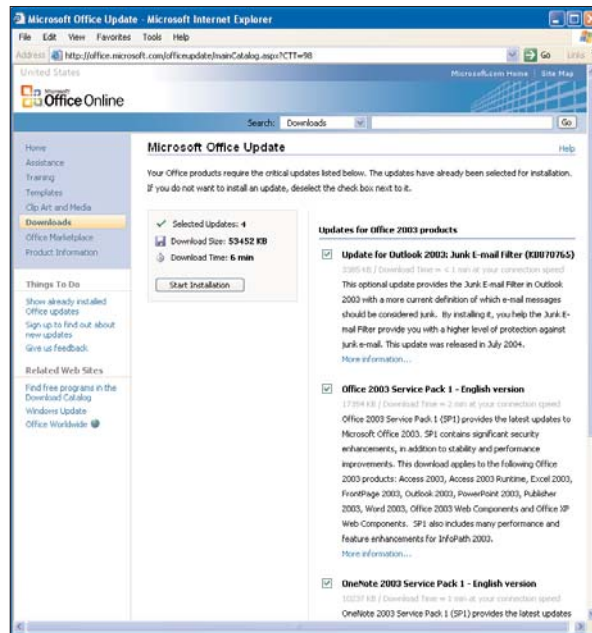
Bookmark this page, then subscribe to the bulletin notification service to ensure notification when Microsoft releases new Security Update bulletins. Also, if you subscribe to a specialized support

program like Premier Support, ask your Technical Account Manager (TAM) to add you to any notifications they send out.

Unfortunately, for now, Microsoft Office uses Office Update, which is a separate update service than Windows Update. For information about patching Office applications visit the Office Update Web site at <http://office.microsoft.com/officeupdate>. This Web site also can scan your computer for missing Office updates, as Figure 1-3 shows.

Figure 1-3

Scanning the Microsoft Office Update Web site for missing updates



Subscribe to the Microsoft newsletter *Inside Office—Product Updates Alert* at <http://www.microsoft.com/office/using/newsletter.asp> to get notified when Microsoft releases a product update including the latest security and performance improvements.

In addition to Microsoft, bookmark other security sites and subscribe to other patch-centric services to keep abreast of newly discovered vulnerabilities and subsequent software updates. Every day these distribution lists send a deluge of information, but keep these messages for at least 30 days. When patch day comes, or if you suspect you have been attacked, you will appreciate the built-up library of technical articles and correspondence.

Don't overlook the Usenet groups, which provide huge and largely unmoderated discussions about most everything including patching. Subscribe to the Microsoft patch and security newsgroups at <http://www.microsoft.com/technet/community/newsgroups/security>. To search other newsgroups for vulnerabilities, use your own provider or a public provider such as Google Groups at <http://groups.google.com>.

8 Keeping Your Business Safe from Attack: Patch Management

Other good third-party notification services for exploits, vulnerabilities, patches, and other security updates include the SecurityFocus Bugtraq at <http://www.securityfocus.com/subscribe?listname=1>, Mitre's Common Vulnerabilities and Exposures at <http://www.cve.mitre.org>, the Carnegie Mellon University CERT at <http://www.cert.org>, the United States Computer Emergency Readiness Team (US-CERT) at <http://www.us-cert.gov>, and the SANS Internet Storm Center at <http://isc.sans.org> among others. Even most antivirus vendors provide links and descriptive information outlining new attacks, vulnerabilities and include links to vendor patches or mitigating steps. For example, check out Symantec at <http://www.sarc.com> and TrendMicro at <http://www.antivirus.com> for detailed information about new viruses and worms and how to prevent them.

Proactive and comprehensive access to new vulnerability and exploit information is essential to making appropriate triage decisions surrounding patching vulnerabilities in your organization. Chapter 2 delves into the contents of Microsoft Security Bulletin Updates in much more detail.

Review All New Security Bulletins with the Team to Assess Risk and Triage Deployment

Now that you have assembled the team and meet regularly, define your process of reviewing new Security Bulletins to assess risk and triage the deployment of new patches. The triage process is important because large companies cannot immediately deploy all patches all the time. You will need to make tradeoff decisions as to when patches will be deployed and how the patching effort will be prioritized with the other work your business conducts.

Although a small company might be able to patch everything right away when a new update is released, a large company hosting complex or mission- and business-critical applications generally does not have this luxury. Updates need testing and deployment in a systematic fashion that reduces the chance that a patch will adversely affect an important system. You never want the cure to be worse than the illness! To intelligently assess new Security Bulletins and their effect on your systems, you must triage each patch. An example of a triage process follows:

Rank the Patch's Applicability to Your Environment.

- Assess the risk if you do not deploy the patch. Generally, you calculate risk as the probability of an event multiplied by the damage that the event could cause. In terms of a patch, the risk might be the chance that someone could compromise the system multiplied by the effect of the break in. Let's use the LSASS DLL as an example again. The risk for this vulnerability is very high because it is easy for an attacker to access the vulnerability through an SSL Web site. And the damage is high because the attacker could take full control of the computer system. High probability times high potential damage equals high risk.
- Assess the damage if someone exploiting the vulnerability that the patch addresses attacks you.
- Assess the patches based on target platform. Microsoft Security Bulletins specify the target of a patch, such as Windows, SQL Server, IE, or Office.
- Determine whether you can make any mitigating efforts in the short-term to shore up your defenses while patch testing occurs.

At the end of this triage assessment, set your sights on determining the criticality and priority for deploying each patch to specific computers in your environment. For example, priority patches likely include immediately exploitable attack vectors such as employees using a vulnerable version of IE to surf infected pages or attackers attempting to infiltrate an unprotected Web server.

Most corporations protect their Internet connections with perimeter firewalls that inspect and permit inbound and outbound network traffic based on ACLs. The use of a perimeter firewall will help mitigate many exploit attack vectors. For example, the RPC exploit required a computer listening on TCP port 135. Most corporate perimeter firewalls ordinarily block this port. Consideration of these mitigating factors when triaging new patches is important, but don't assume that you are always protected. Most firewalls will not protect you from worms or viruses that are distributed through email messages unless those firewalls have built-in antivirus scanning or intrusion prevention capabilities.

When considering your firewall protection, keep the following scenario in mind. Your remote users routinely breach your perimeter firewall by transporting their work laptop from inside your protected LAN to their home, which might be directly connected to the Internet using a DSL or cable connection. Perhaps they are running a base version of SQL Server and Microsoft IIS on their work laptop. They disconnect from the corporate LAN and connect their home computer by plugging directly into their cable modem. Worms that attack IIS and SQL Server (e.g., Nimda, Code Red, SQL Slammer) still plague the Internet and developer's computers run a high probability of being infected. After infection they might either establish a VPN tunnel back into the company or physically carry and connect their laptop onto the company LAN. When reconnected to the LAN and inside the perimeter firewall, infected computers can propagate the worms to other internal systems.

This scenario might affect your triage decision regarding when to deploy a patch to your internal systems. This scenario also provides a good example for implementing system-startup-based and time-based patch management scanning software that routinely checks that patch management status of any system on your LAN. Systems not patched are updated or else quarantined from the network. This practice ensures that even after an initial wave of patch updates, computers brought onto the network later will be patched.

Weigh Deploying Updates vs. Exploit Mitigation Efforts

The triage team also needs to review and recommend mitigating factors for patches, environments, and targets. In the Security Update Bulletins for each patch, Microsoft lists several common mitigating factors specific to that vulnerability. In addition to these, it is important for your triage team to consider factors relevant to your environment. For example, in the IE exploit attack vector described earlier, mitigating factors might be to install a client-based IPSec or perimeter firewall ACL that prohibits outbound Web requests to specific sites. The mitigating action does not necessarily solve the problem but it might buy you time so that patches can be appropriately tested and deployed.

Choosing Software to Deploy Patches

Fundamentally, patching a computer consists of downloading the appropriate software update and executing it on a target computer. Historically, Microsoft product teams introduced distinct patch management technologies. This means that Windows OS updates are very different from Office updates and your patch deployment tools might support one better than the other. (Microsoft is addressing this concern and promises to one day combine all product updates into a common delivery mechanism.)

When configured properly, Automatic Update will check for updates automatically. However, the manual process for deploying patches usually consists of logging onto computers and either visiting Windows Update or manually downloading and installing the appropriate patches. This process is sometimes complicated because Microsoft might release multiple (sometimes three or four) update files per security update depending on the version of software installed. For example, an IE patch

might be released as separate files for IE 5.0, IE 5.5, IE 6.0, etc. This slows the manual process because in a mixed environment you must download each of these versions, then choose the correct patch to run for each computer system you manage. This patch version disparity alone is a compelling reason to purchase and use an effective patch management tool.

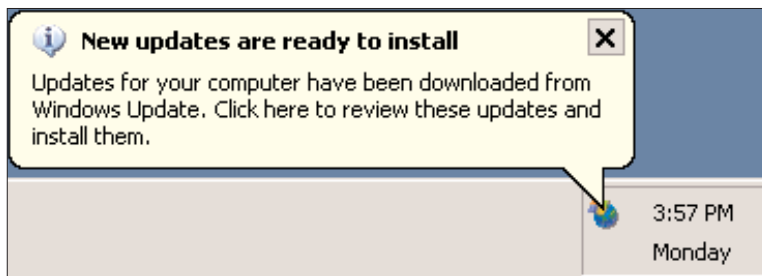
A good patch management tool not only scans a computer for the missing patch, but will also discern the proper version needed, download it, and install it. For example, you can use several tools to scan a set of computers running different software versions, then simply instruct the patch installation software to *deploy patch MS04-xx*. This system ensures the correction version of MS04 is deployed despite the platform. The patch management tool scans the targets, determines the patches necessary, downloads the patches from Microsoft, then installs the correct version on the appropriate systems. Some third-party patch management tools repackage the Microsoft patches into a different format that lets them add features, such as support for multiple (non-Microsoft) software vendors and additional installation functionality. Later this chapter discusses some of the features to watch for when selecting patch management software.

Windows Automatic Updates

Microsoft offers several patch management software packages aimed at different audiences. Small office/home office (SOHO) and individual computer users without a network infrastructure can configure the Windows XP Automatic Updates feature which regularly polls the Microsoft Web site for newly available patches. The Automatic Updates client software identifies the correct patch required for each individual computer and when new patches are available a system tray icon pops up, as Figure 1-4 shows, and notifies the user.

Figure 1-4

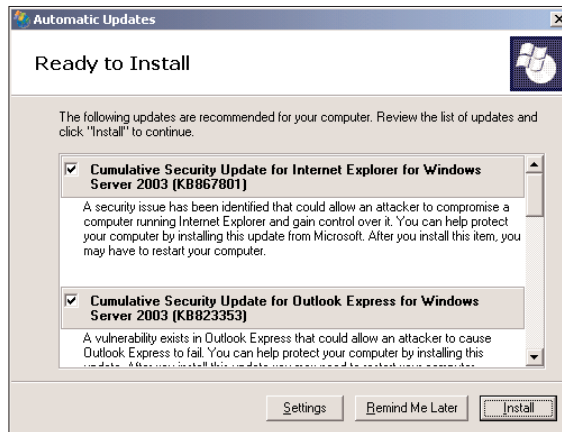
Receiving notification that new updates are ready to be installed



From the Automatic Updates dialog box, the user can review the updates, select updates to install, and automatically install the patch at a specified time, which Figure 1-5 shows.

Figure 1-5

Reviewing and selecting which updates to install



Windows Automatic Update covers patches for a variety of Microsoft products including: Windows, Office, Crystal Reports Web Viewer, Exchange Server, Internet Security and Acceleration Server (ISA Server), MSN Messenger, Virtual PC for Mac, BizTalk Server, Content Management Server (CMS), FrontPage Server Extensions, IIS, SQL Server, and more.

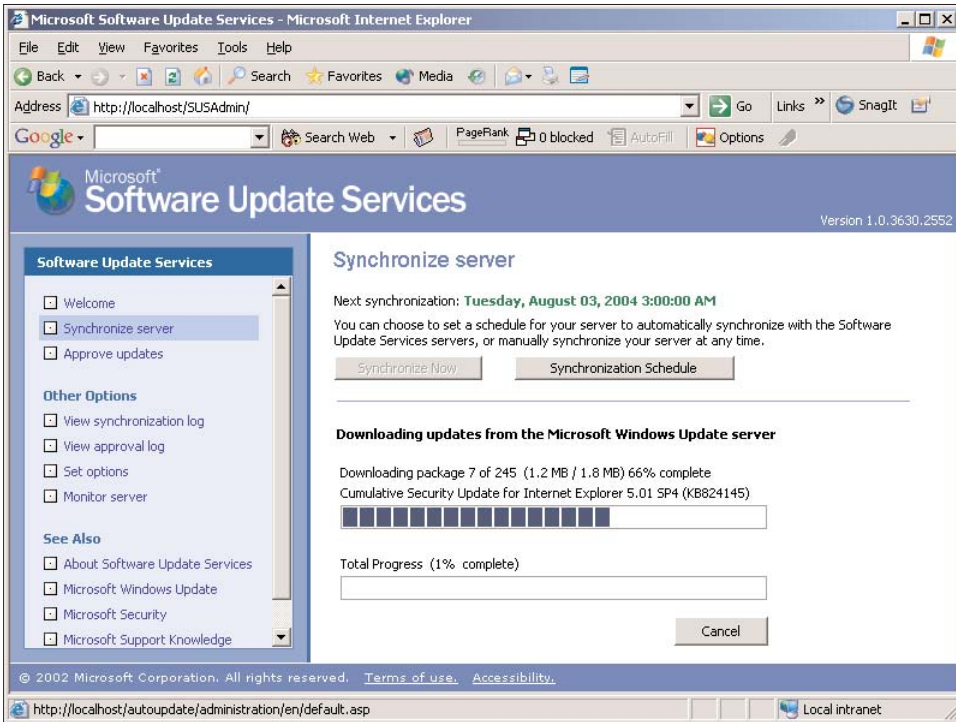
Chapter 2 describes in detail the Microsoft communications. The chapter also contains links to the patches so that you can download them and manually install them on your computer systems.

Microsoft Software Update Services and Windows Update Services

Microsoft also created Software Update Services (SUS) and the soon-to-be-released Windows Update Services (WUS) to provide large companies more control over patch deployment to end user computers. SUS leverages the same client as the previously mentioned Windows Update. This client is included in Windows 2000 SP2 and later and Windows XP SP1 and later releases. But systems using Windows 2000 SP1 or earlier or Windows XP (without SP1 or SP2) need a separate Automatic Update client.

SUS lets you centrally manage the automatic update settings of your end user computers and also lets you deploy your patches from a centralized SUS server in your network. A systems administrator can approve all updates on SUS server and those approved will be sent to the clients. This practice saves WAN bandwidth because not every end user computer needs to repeatedly download the same patches from Microsoft. Instead the SUS server downloads the patches from Microsoft, as Figure 1-6 shows, then each end user's computer downloads the patches from that SUS Server.

Figure 1-6
Downloading updates from a centralized SUS server



After you install SUS inside your corporate network boundaries, it polls the Windows Update server on the Internet for new updates, downloads them, and makes them available for deployment in your corporate environment.

Your central SUS server can also feed other SUS servers located in branch offices, for example for remote deployment to reduce network traffic. Additionally, SUS provides centralized configuration by means of a Group Policy Object (GPO). Configure when and how to download and deploy patches, then assign that GPO to your computers in specified GPO containers such as sites, domains, or OUs. Chapter 6 will cover more details about SUS and the newer WUS.

Microsoft SMS 2003

Microsoft created SMS to help enterprise-size organizations manage a large number of end-user computers. SMS 2003 integrates the patch management features released for SMS 2.0 Feature Pack 1. SMS 2003 provides a much higher degree of targeting and more robust reporting than SUS. For example, you can specify to deploy patches based on machine attributes (e.g., laptops versus desktops) and you also have a fine degree of control over patch deployment. In addition, you can set up a patch deployment package that lets the user choose the most convenient time to install patches within a

3-day window after patch deployment. Chapter 7 explores some of the SMS 2003 features surrounding patch management.

Beyond Microsoft

The software involved in a patch management solution generally scans target systems for missing patches, then deploys patches on those computers. Various software applications add features and functionality to help this process.

Many patch management applications let you create several groups that contain desktops or servers, such as IIS servers, database servers, infrastructure servers. Look for products that ease the process of populating to these groups. For example, can they read Active Directory (AD) to get group or structure information such as domains, sites, or organizational units (OUs)? Can they create groups based on IP address or other characteristics (e.g., software installed) of the target systems? Look for the ability to quickly customize and save patch group memberships. Using predefined groups will save you time during subsequent scanning and deployment procedures.

The patch scanning features vary by product. The most accurate (but frequently slowest) scanning methodologies involve comparing the registry and specific file versions (including size or date) of a target computer with the desired values stored in a patch database. The patch management tool flags a computer when any of the values do not match.

The scan and deployment features also vary by product so be sure to put several products to the test. Some products let you deploy patches immediately following a scan and some let you schedule both the scan and deployment. For example, you can scan anytime to check compliance, then deploy later during specific change windows or at night. Some patch management tools retain a history of scans for auditing purposes or in case a rescan is necessary. Many Microsoft updates require a reboot when installed and different patch management tools let you specify when and how the reboot should occur. Some products use QChain, the Microsoft utility that keeps track of changed files, to minimize multiple reboots through a succession of patch updates. Also check whether the products support Microsoft update rollback features. Not all patches support this feature, but you might find it useful for your patch management software to support patch uninstallation also.

Patching Office products may require the Office installation files. If you want to deploy Office patches, make sure the patch management tool supports Office deployments and check with the vendor to determine whether they support updating multiple versions of Office (each needing separate source files) with a single scan and deploy action.

Installing patches requires administrator access at some level, so make sure the products you select will fit into your user privilege model. For example, will your end users need to be local administrators or does the patch management tool run under a separate privileged account? Some patch management solutions require that a software agent be installed on every computer, yet other solutions scan and deploy entirely from one management console. Agents can provide better feedback and installation control but also increase the software footprint of the computer, which may be an important consideration for server deployments. Agents also tend to provide more robust remote management options and may include basic Quality of Service (QoS) controls, such as bandwidth throttling and checkpoint restarts.

Training

The final essential element to a solid patch management program is to provide quality, comprehensive training to everyone involved with the patch management program. At first consideration you might think of training the systems administrators who use the patch management software day to day. But don't forget about training management who must *buy into* your patch management program and fund the software and resources required to roll out the patches.

Extend your training efforts beyond how to use your patch management software. Include training for the processes behind your entire patch management strategy and tactics. This includes developing documentation and holding meetings regarding the elements presented earlier in this chapter, such as the roles of the various Patch Management Deployment and Triage Team members, how to interpret Microsoft's security software update communications, and how to keep your system inventory current to facilitate patch triage decisions.

When a new exploit ravages the Internet, bring together your patch deployment team and review the exploit's *attack vector* (the method that the exploit used to leverage a particular vulnerability). Discuss how your patching efforts saved (or could have saved) your organization from this exploit. If you were a victim of an exploit resulting from an unpatched vulnerability, immediately conduct a postmortem review. Use this review to play back the steps leading up to the attack. Use the session to help train others affected by the exploit on the importance of your patching processes. Another benefit of a postmortem review immediately following an exploit is that everyone is much more acutely aware of the issues and problems leading up to the exploit and are likely to accept action items for any corrective actions that lead to process improvements. Even if you were not vulnerable to a widespread exploit such as a mass-infecting worm, use the publicity of the event to rally your team to confirm your processes and drill team members with *what if* scenarios to encourage continual process improvement.

Develop training materials that document your patch management process. These materials define the goals of the patch management team and the roles and responsibilities of each team member. For example, a systems administrator might be the point person for installing the patches on specific systems but a developer might be responsible for testing the effect of the patches on the system applications. Clearly document your organization's entire patch management process: from system and application inventory, to patch triage activities, to patch testing, to deployment, and even to follow-up testing. Review with team members their roles in the process and distribute the document for reference. You will find that physically documenting the process helps bring auxiliary team members into your process, which ultimately improves the effectiveness of the entire program.

Training consists of both formal and informal meetings. Formal meetings might include Web-based seminars from your patch management software vendor or in-house expert. Formal training might also include dry-run sessions and drills, which keep staff current and skilled on your chosen patch deployment methodology. Informal training comes in the form of discussion groups or emails that are sure to circulate when preparing for or during a patch management exercise.

Keep up to date on the version and features of your patch management deployment software. This industry is still somewhat new and Microsoft will continue to consolidate and improve its patch update delivery mechanisms. As Microsoft evolves its technologies patch management software vendors will do the same.

Also train Quality Assurance (QA) testers and patch deployment engineers to proficiently use your tools and testing methodologies to ensure that new patches are thoroughly tested and promptly and effectively applied.

Even if you are not a software development company, you might be surprised at the QA resources available to assist with the testing of your patches. Whereas QA testers for software companies test developer's code to look for bugs and performance issues, application service providers (ASPs) use QA staff to test Web sites for proper operation across the target audience of that ASP. Large organizations in more traditional lines of business (LOB) sometimes employ QA testers to test new functionality for enterprise software such as large financial applications, customer relationship management (CRM) systems, point of sale (POS) systems, etc. These people are also commonly experts with the target systems and you will likely find it valuable to tap their knowledge and familiarity with their systems. Plus they might be able to help put together appropriate tests or review your triage decisions to ensure that after a patching exercise the target platform remains fully operational.

Chapter 3 describes ideas and attributes for a patch management testing plan. Ensure that the executors of these testing plans are also familiar with the patching process and methodology. When integrated into the patch management program your organization's QA resources will become your frontline scouts to warn you of any problems that might arise as a result of a particular patch.

The Full Rally

A solid patch management program consists of well-defined processes, effective software, and comprehensive training. Consider developing a Patch Management Deployment and Triage Team to regularly meet and review and prioritize upcoming patches and help marshal the deployment process. In summary, consider these pointers to help set up your patch management program:

- Identify your processes to assess, test, and deploy the updates.
- Create a Patch Management Triage and Deployment Team to help coordinate your patch management activities.
- Subscribe to Microsoft and non-Microsoft patch and security advisories and bulletins. For centralized management, consider subscribing an internal distribution list to the Microsoft Security Bulletins newsletter for distribution within your company.
- Review all new Security Bulletins with the team to assess risk and triage deployment of new patches.
- Weigh deploying updates versus exploit mitigation efforts for different patches, environments, or targets.
- Determine SLAs for different levels of patches, for example, internal versus production or workstation versus server.
- Devise and document testing procedures to ensure that the appropriate groups test and sign off on a patch before released to production. Consider a burn in period when feasible.
- Select patch testing and distribution software effective for your organization and train staff on how to use this software to deploy the updates.
- Scope and cost will often dictate whether to use Windows Update or an external patch management software such as SUS, SMS, or third-party tool to manage the deployment of new updates.
- Drill and train staff not only on the patch management tools but the processes for triaging and testing new software updates.

16 Keeping Your Business Safe from Attack: Patch Management

- Train QA testers to use the same patch management tools and processes as your production teams to ensure consistent testing between labs and production.

Microsoft offers and supports low-cost patch deployment tools and tools that scale for very large enterprises. If Microsoft does not have a solution that fits your organization, consider one of the many new third-party patch management and deployment software packages that have hit the market.

Chapter 2 will examine the Microsoft Update Bulletin and communications. Microsoft uses these primary information delivery mechanisms to inform its customers about newly available patches.