Microsoft®

Solutions for Security
Solutions for Management

The Microsoft Guide to Security Patch Management

patterns & practices

proven practices for predictable results

Table of Contents

Overview

Executive Summary

The Business Problem

Organizations depend on information technology resources and expect them to be trustworthy: a few days of downtime is expensive, while a security compromise of corporate assets can have disastrous consequences.

Viruses and worms such as Klez, Nimda, and SQL Slammer exploit security vulnerabilities in software to attack a computer and launch new attacks on other computers. These vulnerabilities also provide opportunities for attackers to compromise information and assets by denying access to valid users, enabling escalated privileges, and exposing data to unauthorized viewing and tampering.

The operational cost of a day's downtime can be calculated for most, but what if the information with which others entrust your organization is compromised publicly?

A breach of corporate security and the resulting loss of credibility (with customers, partners, and governments) can put the very nature of an organization at risk. Organizations that fail to perform proactive security patch management as part of their information technology security strategy do so at their own peril.

The Response

Microsoft® takes security threats very seriously, quickly providing guidance and, when necessary, security patches for vulnerabilities. The goal at Microsoft is to ensure that customers have the ability to secure their computers from vulnerabilities *before* dangerous and illegal exploits are built into viruses or performed surreptitiously by attackers.

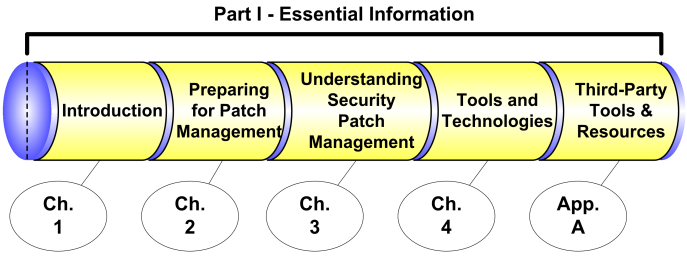
Some organizations regularly apply security patches to mitigate the risk of future attacks and keep their environment secure; many others do not. Staying secure through proactive security patch management is a requirement for keeping information technology resources trustworthy.

The Microsoft Guide to Security Patch Management provides concise information, prescriptive techniques, tools, and templates to help organizations cost-effectively maintain a secure and reliable Microsoft environment through the proactive assessment of vulnerabilities and the application of security patches and appropriate countermeasures.

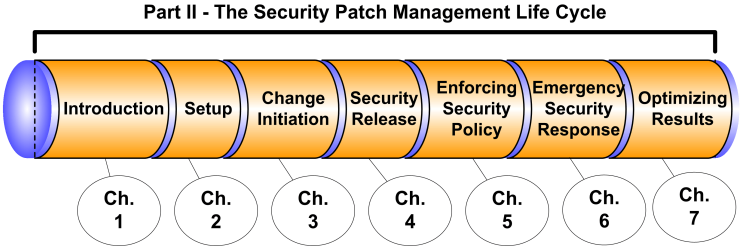
Note: This guide provides information on the maintenance of multiple computers within an organization. If you are responsible for the security and maintenance of your own computer (such as a home user), please keep your computer secure by following Microsoft Security and Privacy Basics: http://www.microsoft.com/security/home/basics.asp.

Reading This Guide

This guide is divided into two parts: the first providing essential security patch management information that is valuable to everyone and the second providing detailed process descriptions and techniques.



Part I: For technology decision makers and professionals who are new to security patch management



Part II: For technology professionals performing the security patch management process

Part I – Essential Information

For information technology (IT) decision makers and technology professionals who are new to security patch management, Part I provides useful information that every organization should understand about how Microsoft approaches software updates and vulnerabilities, plus concise information about processes, tools, techniques, and resources to conduct security patch management effectively. Key concepts, terms, and technologies are introduced in these chapters that should be understood when reading Part II.

1. Introduction to Security Patch Management

This chapter discusses several security issues in the software industry and the resulting impact that they can have on an organization, introduces key terms that are used frequently throughout this guide and identifies some common security vulnerabilities, historical exploits, and the lessons learned from them.

2. Preparing for Patch Management

Chapter 2 introduces the business costs of not doing patch management, and the steps an organization should perform to successfully prepare for proactive patch management. It also identifies key responsibilities that are required throughout the patch management life cycle.

3. Understanding Security Patch Management

This chapter discusses a streamlined process for security patch management, and defines key issues, concepts, and best practices that everyone involved in patch management should understand. This streamlined process also serves as the structural foundation for the prescriptive process and techniques that are presented in Part II. If you will be reading Part II, you do not need to read this chapter.

4. Tools and Technologies

Chapter 4 introduces Microsoft patch assessment and deployment technologies, discusses the costs and capabilities of each, and provides guidance to help an organization decide which patch distribution infrastructure is appropriate for them.

Appendix A: Third-Party Tools and Resources

Appendix A assembles a list of some of the third-party tools and resources that are available to help with security patch management.

Part II – The Security Patch Management Life Cycle

For technology professionals who are performing security patch management in an organization using Microsoft software, Part II discusses the process in more detail while describing tool-specific issues and techniques. The process details are appropriate for any organization, while the issues and techniques focus specifically on the needs of organizations using Microsoft Windows® Update (WU), Microsoft Software Update Services (SUS), or Microsoft Systems Management Server (SMS) for security patch management.

1. Introduction

The first chapter of Part II provides an overview of the security patch management life cycle and introduces techniques. These techniques provide prescriptive technical guidance on the use of various tools to accomplish specific activities throughout the security patch management life cycle.

2. Setup

Even with a patch management infrastructure in place, there are several infrequent activities that are required to support effective security patch management. This chapter discusses these setup activities including configuring and maintaining the patch management infrastructure, inventory identification and baselining the environment, subscribing to security notifications, and establishing ongoing security reports (using tools such as Microsoft Baseline Security Analyzer) to assist with issue identification.

3. Change Initiation

Chapter 3 describes several ongoing maintenance and monitoring activities and how the resulting information is used to respond to security issues. These activities include regularly reviewing Web sites, security notifications, and security reports to identify new software updates and security issues, determining their relevance in your environment, downloading and quarantining new software updates for use in subsequent steps, initiating a release, and other typical responses to security issues.

4. Security Release

Releasing a software update or related countermeasures is the typical response to a newly-identified vulnerability. This chapter discusses the aspects of change management, release management (including software update testing), and change review (possibly including rollback) that a security release should follow.

5. Enforcing Security Policy

New computer installations, lab equipment, mobile users, and decentralized administration can all be sources of previously addressed vulnerabilities recurring in your environment. Recurring vulnerabilities are at increased risk of exploitation by viruses, worms, and attack tools that remotely scan computers for security weaknesses and published vulnerabilities. Chapter 5 discusses some strategies that organizations can use to eliminate old vulnerabilities if they recur.

6. Emergency Security Response

This chapter describes how to prepare for an emergency caused by exploited security vulnerabilities and the critical information, steps, and best practices that are necessary to respond effectively if your organization's information technology is at risk or under attack.

7. Optimizing Results

Occasionally, it is important to review how effectively your organization performs security patch management and how well you keep the information technology environment trustworthy. Chapter 7 discusses the key performance indicators that can be measured and improved over time, as well as a few resources that are available to help improve performance.

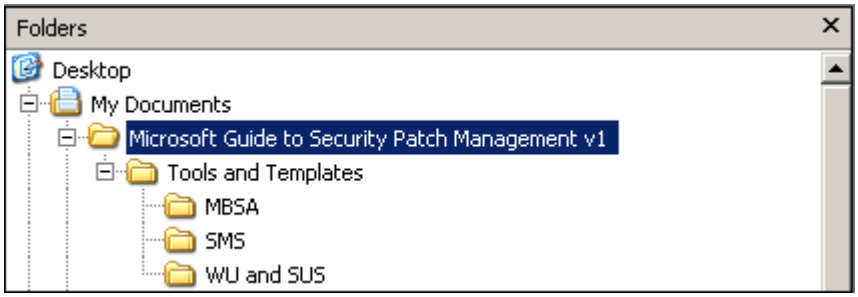
Downloads and Resources

Download this guide with included Tools and Templates:
http://go.microsoft.com/fwlink/?LinkId=16286

Accessing the Guide Tools and Templates

Tools and Templates provided with the downloadable version of this guide include text-based scripts, queries, documents, and forms related to security patch management. No executable programs (compiled code) are provided in the download package.

After downloading the code-signed, self-extracting package entitled Microsoft_Guide_To_Security_Patch_Management_v1.exe, you can extract the file on your hard drive. The resulting directory structure will look similar to the following, with tools and templates contained in the Tools and Templates directory and appropriate sub-directories:



The Readme.txt file (available on the download page and in the download package) includes a full list of all the files in the download package.

Related Resources

- Microsoft Security Resources on TechNet:
  http://www.microsoft.com/technet/security.
- Microsoft Security Technologies Community Center:
  http://www.microsoft.com/communities/security.

Scope of This Guide

Keeping your Microsoft environment secure and reliable is a priority for Microsoft. While all software updates can be identified and deployed through the process described in this guide, the detailed techniques and best practices focus primarily on security patches. Failure to keep up to date on security patches (and the prerequisite service packs) can have a devastating effect on organizations when that vulnerability is exploited.

Organizations that use Microsoft tools for security patch management will find the techniques valuable, whereas organizations using any technologies for patch management can learn from the detailed process and policy information provided.

Note: Information about Microsoft futures is kept to a minimum in this guide. If you are interested in patch management futures and the improvements and commitments that Microsoft is making in this area, see the Improving Patch Management white paper: http://www.microsoft.com/security/whitepapers/patch_management.asp.

Microsoft Operations Framework (MOF) and Microsoft Solutions for Management (MSM)

This guide was created by distilling information provided in the Microsoft Solutions for Management (MSM) patch management guides and providing an additional focus on security. MSM provides best practices, best practice implementation services, and best practice automation to help organizations achieve operational excellence.

This guide is consistent with the terminology of the Microsoft Operations Framework (MOF): operational guidance that enables organizations to achieve mission-critical system reliability, availability, supportability, and manageability of Microsoft products and technologies.

Readers are not expected to understand or have implemented MOF or MSM to read or use this guide. Several aspects of MSM and MOF have been simplified in this guide to enable broad access and adoption.

For more information on MSM and MOF, see:

- Microsoft Solutions for Management: http://www.microsoft.com/technet/itsolutions/msm.
- Microsoft Operations Framework: http://www.microsoft.com/mof.

Out of Scope

For the purposes of this guide, Microsoft applications built for operating systems other than Windows (such as the Macintosh platform) and embedded software provided on hardware sold by original equipment manufacturers (OEMs) are not discussed. OEM software is typically serviced by the hardware manufacturer.

There are several security topics that are closely related to security patch management that are introduced and referenced in this guide, but not described in detail. Where available, this guide directs the reader to existing resources for the following activities:

- **Anti-virus scanning.** Viruses are common attacks that exploit software vulnerabilities.
- **Intrusion detection.** The discovery of viruses, worms, and attackers in your environment can be facilitated by intrusion detection solutions.
- **Operating system and application hardening.** Many attacks exploit weak security configurations.

The focus of this guide is on performing security patch management. Guidance on the architecture and deployment of a software or patch distribution infrastructure containing SUS, SMS, or third-party tools is not provided. However, existing resources on these topics are referenced.

Giving Feedback

We want your feedback on this material. In particular, we would be grateful for any guidance on the following topics:

- How useful was the information that we provided?
- Were the step-by-step procedures accurate?
- Were the chapters readable and interesting?
- What other specific techniques would you like to see?
- Overall, how would you rate the guidance?

Send your feedback to the following e-mail address: secwish@microsoft.com. We look forward to hearing from you.

Consulting and Support Services

There are many services available to assist organizations in their patch management efforts. The following links are a great start for locating the services that you need:

- Search the Microsoft Resource Directory for Microsoft Gold Certified Partners, Microsoft Certified Technical Education Centers, Microsoft Certified Partners, and products from independent software vendors that use Microsoft technologies.
- Find the Microsoft Services (Consulting and Support services) that are appropriate for your organization.

Acknowledgments

Microsoft Solutions for Security (MSS) would like to acknowledge and thank the team that produced the *Microsoft Guide to Security Patch Management*. The following people were directly responsible or made a substantial contribution to writing and reviewing this guide.

Development Team

Authors

José Maldonado, Microsoft Services

Rod Trent, Studio B

Testers

Greg Feiges, Microsoft Services

Mike Jenne, Microsoft Services

Editor

Jennifer Kerns, Wadeware

Program Manager

Derick Campbell, MSS

Anthony Baron and Graham Stenson of the Microsoft Solutions for Management (MSM) team also provided significant contributions and advice.

Beta Reviewers

Microsoft Reviewers

Kristie B. R. Atwood, Enterprise Technical Sales

Gigel Avram, Software Update Services

David H. Baur, Microsoft Services

Cory Delamarter, Operations and Technology Group

Chris Geier, Microsoft Services

Robert Hensing, PSS Security

Maxim Kapteijns, Microsoft Services

David Lef, Operations and Technology Group

Patrick Martin, Microsoft Services

Eric Price, Windows Sustained Engineering

Christopher Reinhold, Microsoft Services

John Roller, Enterprise Technical Sales

Fernando Pessoa Sousa, Microsoft Services

Bill Stackpole, Microsoft Services

External Reviewers

Susan E. Bradley, Microsoft MVP

Dean Farrington, Wells Fargo

Nina Ferguson, Qwest

Adam S. Greene, Intel

Glenn Hellriegel, Verizon Wireless

Paul Hudson, Attenda Ltd.

Deanna Jenness, First Call Computer Solutions

Russ Klanke, Verizon Wireless

Jeff Middleton, Microsoft MVP

Hans Muller, Attenda Ltd.

Chad Sharp, Intel

Kerry Steele, Citadel Security Software

Maria M. Tsiolakki, University of Cyprus

Microsoft would also like to thank the National Institute of Standards and Technology for their invaluable input and participation in the beta review of this guide.

Release Notes

This is the v1.1 release of the Microsoft Guide to Security Patch Management, which was completed on Thursday July 3rd, 2003.

Changes in This Version

Version 1.1

Two corrections have been made:

- **Page 59—Office Update support**. Part I, Chapter 4, "Tools and Technologies." Updated the sentence to correctly indicate that Office Update works for Office 2000 and above, not Office 97 and above as previously indicated.
- **Page 115—Ohotfix.exe description**. Part II, Chapter 4, "Security Release." Corrected the description of Ohotfix.exe to indicate that it is a launcher for MSP (Windows Installer patch) files, not a custom version of the Windows Installer as previously indicated.

Minor update on July 24th, 2003:

- The comments in MBSAScan.wsf.txt were causing it to fail so it has been replaced with a corrected version.
- Cover page changed to reflect the patterns and practices branding.

Version 1.0

This was the first version of this guide, released on Monday June 30th, 2003.

Unresolved Issues and Resolutions

FWLink on TechNet

Several links within this guide use a forwarding mechanism on Microsoft.com called FWLink. Some computers occasionally receive an Object Not Found message when a forward link points to a TechNet location. To resolve this issue, click **Refresh** on your browser toolbar.

Part I

Essential Information

1

Introduction to Security Patch Management

Secure IT Management and Operations

Computer security has become a critical element of managing technology investments.

Implementing, executing, and continually improving computer security is increasingly important as technology evolves and attackers develop new methods to exploit security vulnerabilities and negatively impact business operations.

Secure information technology (IT) management and operations, including security patch management, is the primary line of defense available to organizations that are interested in protecting themselves from these threats.

Security patch management, operating system and application hardening, proactive virus detection, intrusion detection, and regular review of security settings and account permissions are all crucial elements of secure IT management and operations. Each is part of an effective defense in-depth strategy that is *required* to reduce an organization's exposure to computer crime today.

The Cost of Weak Security

It is difficult to quantify the cost of security breaches because most companies do not report attacks. However, the Computer Security Institute and the U.S. Federal Bureau of Investigation perform an annual computer crime and security survey that tallied more than $201 million in quantified financial losses in 2002. Among respondents, the most frequently cited forms of attack were viruses (82 percent) and insider abuse of network access (80 percent). Theft of proprietary information caused the greatest financial loss, with an average reported loss of $2.7 million.

Note: To access the 2003 CSI/FBI Computer Crime and Security Survey, see: http://www.gocsi.com/db_area/pdfs/fbi/FBI2003.pdf

---

The consequences of criminal attacks on your organization can be severe—resulting in damaged data and assets, business interruption, and infiltration and access to confidential and classified resources. After a computer is infiltrated, applying the security patch is no longer a sufficient remedy to guarantee its security—successfully recovering from an attack may require the complete reinstallation of *every* compromised asset.

Security Patch Management

Security patch management is a necessary process on all platforms—every major software vendor that is committed to security will release security patches in response to newly identified vulnerabilities. There is no widely used operating system or application that is immune from attackers who spend their time trying to locate vulnerabilities to exploit.

The term *patch management* describes the tools, utilities, and processes for keeping computers up to date with new software updates that are developed after a software product is released. *Security patch management* is a term used throughout this guide that is intended to describe patch management with a focus on reducing security vulnerabilities.

Proactive security patch management is a requirement for keeping your technology environment secure and reliable. As part of maintaining a secure environment, organizations should have a process for identifying security vulnerabilities and responding quickly. This involves applying software updates, configuration changes, and countermeasures to eliminate vulnerabilities from the environment and mitigate the risk of computers being attacked. The nature of many attacks requires only a single vulnerable computer on your network, so this process should be as comprehensive as possible.

The majority of successful attacks come from the exploitation of only a few software vulnerabilities. This trend can be attributed to opportunistic attackers who take the easiest and most convenient routes, and exploit the best-known flaws by using the most effective and widely-available attack tools. Such attackers count on organizations not fixing known problems and they often attack indiscriminately, scanning the Internet for vulnerable computers. Attackers do not generally find the original vulnerability, but instead find the code to exploit it. It does not take a security expert to exploit a vulnerability and attack others.

Security Patch Management and IT Operations

Security patch management should be considered a subset of a larger change and release management process. The implementation of security patch management is best achieved when it is a consistent and integral part of an organization's standard operational processes. Without operational consistency, a separate process for security patch management can increase the overall cost of ownership and will introduce unnecessary ambiguity in the organization.

This guide reinforces this operational consistency by leveraging the processes and terminology of the Microsoft® Operations Framework (MOF). MOF provides prescriptive guidance on change and release management and many other service management functions.

Security Terminology

This section introduces key terminology that people should understand when participating in the security patch management process.

The following table describes several security terms that are used throughout this guide.

Table 1.1: Important Security Terms

Term	Definition
Vulnerability	A software, hardware, procedural weakness, feature, or configuration that could be a weak point exploited during an attack. Also called an exposure.
Attack	A threat agent attempting to take advantage of vulnerabilities for unwelcome purposes.
Countermeasure	Software configurations, hardware, or procedures that reduce risk in a computer environment. Also called a safeguard or mitigation.
Threat	A source of danger.
Threat agent	The person or process attacking a system through a vulnerability in a way that violates your security policy.

Vulnerabilities

The following table lists several typical software vulnerabilities.

Table 1.2: Vulnerabilities

Term	Definition
Buffer overrun	An unchecked buffer in a program that can overwrite the program code with new data. If the program code is overwritten with new executable code, the effect is to change the program's operation as dictated by the attacker.
Privilege elevation	Allows users or attackers to attain higher privileges in certain circumstances.
Validation flaw	Allows malformed data to have unintended consequences.

MSRC Vulnerability Severity Ratings

The Microsoft Security Response Center (MSRC) uses severity ratings to help organizations determine the urgency of vulnerabilities and related software updates.

Table 1.3: Vulnerability Severity Ratings

Rating	Definition
Critical	A vulnerability whose exploitation could allow the propagation of an Internet worm without user action.
Important	A vulnerability whose exploitation could result in compromise of the confidentiality, integrity, or availability of users' data, or of the integrity or availability of processing resources.
Moderate	Exploitability is mitigated to a significant degree by factors such as default configuration, auditing, or difficulty of exploitation.
Low	A vulnerability whose exploitation is extremely difficult, or whose impact is minimal.

For more information about MSRC vulnerability severity ratings, see the Microsoft Security Response Center Security Bulletin Severity Rating System: http://www.microsoft.com/technet/security/policy/rating.asp.

Threat Categories

Microsoft developed the STRIDE model to categorize software threats. These categories are often used in Microsoft security bulletins to describe the nature of a security vulnerability.

Table 1.4: STRIDE Model of Threat Categories

Term	Definition
Spoofing identity	Illegally obtaining access and use of another person's authentication information, such as a user name or password.
Tampering with data	The malicious modification of data.
Repudiation	Associated with users who deny performing an action, yet there is no way to prove otherwise. *Nonrepudiation* refers to the ability of a system to counter repudiation threats (such as signing for a received parcel so that the signed receipt can be used as evidence).
Information disclosure	The exposure of information to individuals who are not supposed to have access to it, such as accessing files without having the appropriate rights.
Denial of service	An explicit attempt to prevent legitimate users from using a service or system.
Elevation of privilege	Where an unprivileged user gains privileged access. An example of privilege elevation would be an unprivileged user who contrives a way to be added to the Administrators group.

Note: For more information about the STRIDE model and how Microsoft trains developers to write secure code, see Howard, Michael and David LeBlanc, *Writing Secure Code, Second Edition*, Redmond, WA: Microsoft Press, 2002. http://www.microsoft.com/mspress/books/5957.asp.

Threat Agents

Malicious threats are attacks from inside and outside a network that have the intent to harm or disrupt an organization. Non-malicious threats usually come from untrained employees who are unaware of security threats and vulnerabilities.

The following table describes several malicious threat agents.

Table 1.5: Threat Agents

Term	Definition
Virus	An intrusive program that infects computer files by inserting copies of self-replicating code and deleting critical files, makes system modifications, or performs some other action to cause harm to data on the computer or to the computer itself. A virus attaches itself to a host program.
Worm	A self-replicating program, often malicious like a virus, that can spread from computer to computer without infecting files first.
Trojan horse	Software or e-mail that professes to be useful and benign, but which actually performs some destructive purpose or provides access to an attacker.
Mail bomb	A malicious e-mail sent to an unsuspecting recipient. When the recipient opens the e-mail or runs the program, the mail bomb performs some malicious action on their computer.
Attacker	A person or organization carrying out an attack.

Note: While automated threats such as viruses are written to take advantage of specific vulnerabilities, an attacker who is targeting your organization has no such limitations—an attacker tries to compromise an environment by any means available.

Directed attacks can be carried out locally or remotely and can include an exhaustive search for one of many possible vulnerabilities including software vulnerabilities, weak passwords, weak security configurations, and security policy or training vulnerabilities.

How Microsoft Fixes Software After Release

Microsoft is committed to protecting customers from security vulnerabilities. As part of this effort, Microsoft makes available periodic releases of software updates. For more information on this effort, see the Trustworthy Computing white paper: http://www.microsoft.com/presspass/exec/craig/10-02trustworthywp.asp.

Every Microsoft product group includes a sustaining engineering team that develops software updates for problems that are discovered after the product has been released.

When Microsoft is made aware of a security vulnerability, the issue is evaluated and verified by the MSRC and the appropriate product groups. The product group's sustaining engineering team then creates and tests a security patch to remedy the issue, while the MSRC works with the reporter of the vulnerability to coordinate the release of public information in the form of a security bulletin that has the security patch details.

The software update is then distributed through the Microsoft Download Center and other services, such as Microsoft Windows® Update, Microsoft Office Update, Microsoft Software Update Services (SUS), and Microsoft Systems Management Server (SMS) with the SUS Feature Pack.

Just as the software update is about to release, the MSRC sends out a related security bulletin.

Note: Security patches are developed for multiple versions of the operating system and applications. To understand the support levels that you can expect for different software versions, you can review the Microsoft product support life cycle policies at: http://support.microsoft.com/default.aspx?scid=fh;[LN];lifecycle.

Typically, security patches are made available for supported products on the current service pack and one previous. However, this is not always the case: check the product support life cycle policies for your products to be sure.

Software Update Terminology

The following table lists the new Microsoft standard terms for software updates, effective June 30, 2003. Note that the term *patch* is no longer used by Microsoft to describe a software update, except as part of the term *security patch* or when describing the process of *patch management*, which is a well understood term in the software industry.

Table 1.6: New Microsoft Terminology for Software Updates

Term	Definition
Security patch	A broadly released fix for a specific product addressing a security vulnerability. A security patch is often described as having a *severity*, which actually refers to the MSRC severity rating of the vulnerability that the security patch addresses.
Critical update	A broadly released fix for a specific problem addressing a critical, non-security related bug.
Update	A broadly released fix for a specific problem addressing a non-critical, non-security related bug.
Hotfix	A single package composed of one or more files used to address a problem in a product. Hotfixes address a specific customer situation, are only available through a support relationship with Microsoft, and may not be distributed outside the customer organization without written legal consent from Microsoft. The terms QFE (Quick Fix Engineering update), patch, and update have been used in the past as synonyms for hotfix.
Update rollup	A collection of security patches, critical updates, updates, and hotfixes released as a cumulative offering or targeted at a single product component, such as Microsoft Internet Information Services (IIS) or Microsoft Internet Explorer. Allows for easier deployment of multiple software updates.
Service pack	A cumulative set of hotfixes, security patches, critical updates, and updates since the release of the product, including many resolved problems that have not been made available through any other software updates. Service packs may also contain a limited number of customer-requested design changes or features. Service packs are broadly distributed and tested by Microsoft more than any other software updates.
Integrated service pack	The combination of a product with a service pack in one package.
Feature pack	A new feature release for a product that adds functionality. Usually rolled into the product at the next release.

Note: Because these definitions are new, several existing resources and tools do not use them as they are defined in the previous table.

The Importance of Proactive Security Patch Management

There have been several widely-publicized attacks and vulnerabilities related to Microsoft software. Many organizations with proactive security patch management in place were not impacted by these attacks, because they acted on information that Microsoft made available in advance of the attack.

In the following table, several historical attacks are identified, along with the date of the attack. In each case, an MSRC bulletin had previously been released that identified the vulnerability and described how to prevent future exploits of it (through software updates and other countermeasures). The last column in the table, *Days Available Before Attack*, lists the number of days that organizations had to implement the MSRC recommendations and avoid the future attack.

Table 1.7: Historical Attack Examples and Related MSRC Bulletins

Attack Name	Date Publicly Discovered	MSRC Severity	MSRC Bulletin	MSRC Bulletin Date	Days Available Before Attack
Trojan.Kaht	5-May-03	Critical	MS03-007	17-Mar-03	49
SQL Slammer	24-Jan-03	Critical	MS02-039	24-Jul-02	184
Klez-E	17-Jan-02	*	MS01-020	29-Mar-01	294
Nimda	18-Sept-01	*	MS00-078	17-Oct-00	336
Code Red	16-Jul-01	*	MS01-033	18-Jun-01	28

* Bulletins released before MSRC severities in place

This guide was written to help organizations prevent future attacks like these, specifically focusing on the *Days Available Before Attack* column in the table. Many organizations successfully avoided the attacks listed in the table through proactive security patch management.

Note: The preceding table does not capture directed, intentional attacks performed by people inside or outside the target organization, who searched for and exploited security vulnerabilities with criminal intent. Proactive security patch management is an effective way to limit attacks that target known software vulnerabilities.

To provide a better understanding of the relationship between MSRC bulletins and the opportunities they give to organizations that want a secure environment, the following sections briefly describe two historical attacks: the Code Red and SQL Slammer worms.

Avoiding Attacks, Example 1: Code Red

Code Red is a worm that spread very quickly and had the potential for great impact. On July 16th, 2001 the original Code Red worm spread to 250,000 computers in only nine hours. Impact of the worm included slowed Internet speed, Web page outages and defacements, and disruption of business and personal applications such as e-mail and e-commerce.

Code Red exploited a buffer overrun vulnerability within IIS to execute code on Web servers. IIS is installed by default with Microsoft Windows Server 2000 and is used by many applications.

Some organizations avoided Code Red by following the directions of MS01-033, an MSRC security bulletin released on June 18, 2001, 28 days before Code Red was released.

For more information on this security bulletin, including technical aspects and countermeasures, see:
http://www.microsoft.com/technet/security/bulletin/MS01-033.asp.

Avoiding Attacks, Example 2: SQL Slammer

SQL Slammer (or Sapphire) is a worm that targets Microsoft SQL Server™ 2000 and Microsoft Data Engine (MSDE) 2000 systems, resulting in a high volume of network traffic on both the Internet and private internal networks, acting (some might say unintentionally) as an effective denial of service attack.

At approximately 9:30 P.M. PST on Friday, January 24, 2003, SQL Slammer caused a dramatic increase in network traffic worldwide. An analysis of the SQL Slammer worm shows:

- The worm required roughly 10 minutes to spread worldwide, making it by far the fastest worm to date.
- In the early stages, the number of compromised hosts doubled in size every 8.5 seconds.
- At its peak, (achieved approximately three minutes after the worm was released), it scanned the net at over 55 million IP addresses per second.
- It infected at least 75,000 victims and probably considerably more.

SQL Slammer exploited a buffer overrun vulnerability, which was first identified by Microsoft in security bulletin MS02-039 (July 2002), 184 days before the attack, and was identified again in security bulletin MS02-061. With each bulletin, a security patch was offered as well as appropriate countermeasures.

For more information on this security bulletin, including technical aspects and countermeasures, see:
http://www.microsoft.com/technet/security/bulletin/MS02-039.asp.

Lessons Learned from SQL Slammer

One of the challenges organizations faced in avoiding SQL Slammer was the ubiquitous nature of MSDE and even SQL Server, because they are installed and used by many other products.

The SQL Slammer attack highlighted three important lessons on the nature of security vulnerabilities:

- Having an accurate sense of all the computers, products, and technologies that are present in your environment is an important prerequisite for successful security patch management.
- An effective attack does not require vulnerabilities on high-value assets. SQL Slammer effectively interrupted mission-critical operations even through low-value, vulnerable computers on the same network.
- Deploying a security patch once may not be sufficient to eliminate a vulnerability. Regular scanning to identify the recurrence of vulnerabilities coupled with incident management to address them is equally important.

Note: SQL Slammer also taught Microsoft many lessons about how Microsoft needs to improve patch management tools, software update quality, external communications, and internal sustaining engineering processes. For more information on the efforts Microsoft is making in this area, see the Improving Patch Management white paper: http://www.microsoft.com/security/whitepapers/patch_management.asp.

2

Preparing for Patch Management

To prepare for patch management, it is essential to fully understand the business importance of patch management for your specific environment and the technologies and skills that you have (or don't have) to perform proactive patch management. Next, teams and responsibilities can be assigned to ensure patch management is carried out effectively, as part of normal operations. Successful patch management, like security and operations, is achieved through a combination of people, processes, and technology.

Evaluate Your Environment, Risks, and Needs

To determine how much effort to put into security patch management, first assess the impact of poor (or reactive) patch management on the business, then determine if the organization's capabilities and infrastructure are sufficient to perform patch management effectively. Use this information to decide what your organization's goals for proactive security patch management should be.

This section defines these evaluative activities at a high level.

The Business Problem

The business problem should capture and summarize the business impact of not having patch management in place, and thus strictly reacting to security problems when they occur. This is best accomplished by showing the financial impact of successful attacks such as viruses and worms (as well as targeted attacks instigated by perpetrators inside and outside your organization), as well as the negative impact on any related business goals and objectives.

The following are some areas to consider when determining the potential financial impact of poor patch management:

- **Downtime.** What is the cost of computer downtime in your environment? What if critical business systems are interrupted? Determine the opportunity cost of lost end-user productivity, missing transactions on critical systems, and lost business during an incident. Downtime is caused by most attacks, either by the attack itself or the corresponding remediation required when recovering. Historical attacks have left computers down for several days.
- **Remediation time.** What is the cost of fixing a wide-ranging problem in your environment? How much does it cost to reinstall a computer? What if you had to reinstall all your computers? Many security attacks require a complete reinstallation to be certain that back doors (permitting future exploits) were not left by the attack.
- **Questionable data integrity.** In the event that an attack damages data integrity, what is the cost of recovering that data from the last known good backup, or confirming data correctness with customers and partners?
- **Lost credibility.** What does it cost if you lose credibility with your customers? How much does it cost if you lose one or more customers?
- **Negative public relations.** What is the impact to your organization from negative public relations? How much could your stock price or company valuation fall if you are seen as an unreliable company to do business with? What would be the impact of failing to protect your customer's personal information, such as credit card numbers?
- **Legal defenses.** What might it cost to defend the organization from others taking legal action after an attack? Organizations providing important services to others have had their patch management process (or lack of one) put on trial.
- **Stolen intellectual property.** What is the cost if any of your organization's intellectual property is stolen or destroyed?
- **Other areas.** What could the cost of forensic investigations, coordinating with law enforcement, and taking legal action against attackers be?

Use past attacks to help determine these costs, but keep in mind that most viral attacks experienced historically haven't damaged computer systems as significantly as they could have. Targeted attacks by attackers can also be very costly to your organization.

The Assessment

The assessment phase should summarize the tools and assets that are available to participate in patch management, and the capabilities of the organization to perform patch management.

Some of the critical areas to evaluate are:

- **Operating systems and versions.** How many different operating systems are supported? Having different operating systems and versions makes it more challenging to perform comprehensive patch management.
- **Software applications and versions.** How many different software applications and versions might need security patches? Are all of the applications currently supported? How many of them have had security patches issued in the last year? Multiple versions of the same applications can make patch management more challenging.
- **Computer inventory and critical systems assessment.** How many computers that must be managed exist in the environment? Which systems are most critical to company operations, requiring high availability? Is an inventory of computers and software maintained? Is the inventory maintained manually, or automated through a tool? An up-to-date inventory of computers, operating systems, and applications is crucial for patch management.
- **Unmanaged computers.** How many computers that are not centrally configured or managed exist in the environment? These assets still need to be addressed by security patch management, but the nature of distributed administration can create additional complexity when eliminating vulnerabilities.
- **Current vulnerability status.** Are computers kept up to date with the latest service packs and security patches? Is there a good security policy in place that defines secure computer settings and standards for the organization? How are vulnerabilities discovered in the environment? Regular vulnerability scanning, using tools such as the Microsoft Baseline Security Analyzer, are important for ensuring that vulnerabilities are identified.
- **Network infrastructure.** Understand the layout of your network infrastructure, its capabilities, and its security level. Is the network protected from common threats? Are firewalls in place? Are wireless networks secured?
- **Software distribution.** Is a software distribution infrastructure in place? Can it be used to distribute software updates? Does it service all computers in your environment?
- **People and skills.** Are there enough skilled people to perform security patch management? Are people aware that patch management is necessary? Do they understand security settings, common computer vulnerabilities, software distribution techniques, remote administration, and the patch management process?
- **Operational effectiveness.** Are there standard operations processes in place, or are day-to-day operations largely unstated and imprecise? Do processes exist for change management and release management, even informal ones? Securing an environment from attack should not be left to chance or ad hoc operations.

Ensure Sponsorship and Determine Goals

Security programs such as security patch management are more easily accomplished with executive support. Technology professionals can drive security patch management infrastructure, tools, techniques, and processes, but without executive sponsorship it will be difficult to ensure sufficient resources and broad support for areas such as security policy, which must be applied across the organization to be truly effective.

Clear business goals and supporting objectives should be created and understood by the sponsoring executives. The business goals should cover each of the following areas:

- Goal: Minimize the occurrence and severity of attacks.
  - Ensure the proactive assessment and installation of all necessary software updates.
  - Ensure a security policy exists, including standard computer security settings, that are consistent with the level of risk the organization is willing to assume. (No security policy = maximum risk.)
  - Ensure minimal vulnerabilities and security policy conformance through regular vulnerability scanning of the environment.
  - Ensure rapid response time to attacks through proactive intrusion detection.
  - Ensure the organization has tools, skills, and an incident response plan in place to effectively combat attacks should they occur.
- Goal: Optimize the time and resources spent on security patch management.
  - Use tools that help automate vulnerability scanning, inventory and patch assessment, and patch deployment.
  - Put processes in place to ensure that security patch management becomes an effective part of normal operations.

Part II of this guide provides details and resources to help achieve each of these goals.

Establish Teams and Responsibilities

Depending on an organization's needs and size, patch management might be performed by a single person or a whole team of people working in a central or decentralized manner. To ensure success, responsibilities should be well-defined and team members should be held accountable for results.

To assist with operational team structure and responsibilities, the Microsoft Operations Framework (MOF) team model includes six role clusters that identify the duties that need to be performed in an effective operations environment.

Role clusters can be used to define entire operations teams, if you want. In smaller environments, a few people may take on multiple roles.

For full descriptions of the MOF role clusters and how they relate to various operations processes, see the MOF Team Model for Operations white paper: http://www.microsoft.com/technet/itsolutions/tandp/opex/mofrl/MOFTMl.asp.

Next Steps

After evaluating what your organization needs and ensuring that the appropriate roles and responsibilities are in place, you next need to select and implement the required infrastructure so that security patch management can become an integral part of IT operations.

Implementation and Operations

Implementation focuses on selecting and implementing the necessary tools, technologies, and infrastructure that will assist operations with effectively accomplishing the goals established for security patch management.

Part I, Chapter 4, "Tools and Technologies" and Part I, Appendix A, "Third-Party Tools and Resources" provide information on the tools that are available for automating vulnerability scanning, inventory and patch assessment, patch deployment, and some additional security activities. Part I, Chapter 4 also includes guidance on choosing a Microsoft software update distribution infrastructure.

Note: For guidance and proven practices for planning, building, and deploying IT solutions successfully, see the Microsoft Solutions Framework at: http://www.microsoft.com/msf.

Security patch management operations is the primary focus of this guide, and are described at a high level in Part I, Chapter 3, "Understanding Security Patch Management" and in detail in all of Part II.

3

Understanding Security Patch Management

Note: Part II of this guide elaborates on the same life cycle that is introduced in this chapter, providing detailed practices and techniques for several technologies. If you will be reading Part II of this guide, it is not necessary to read this chapter.

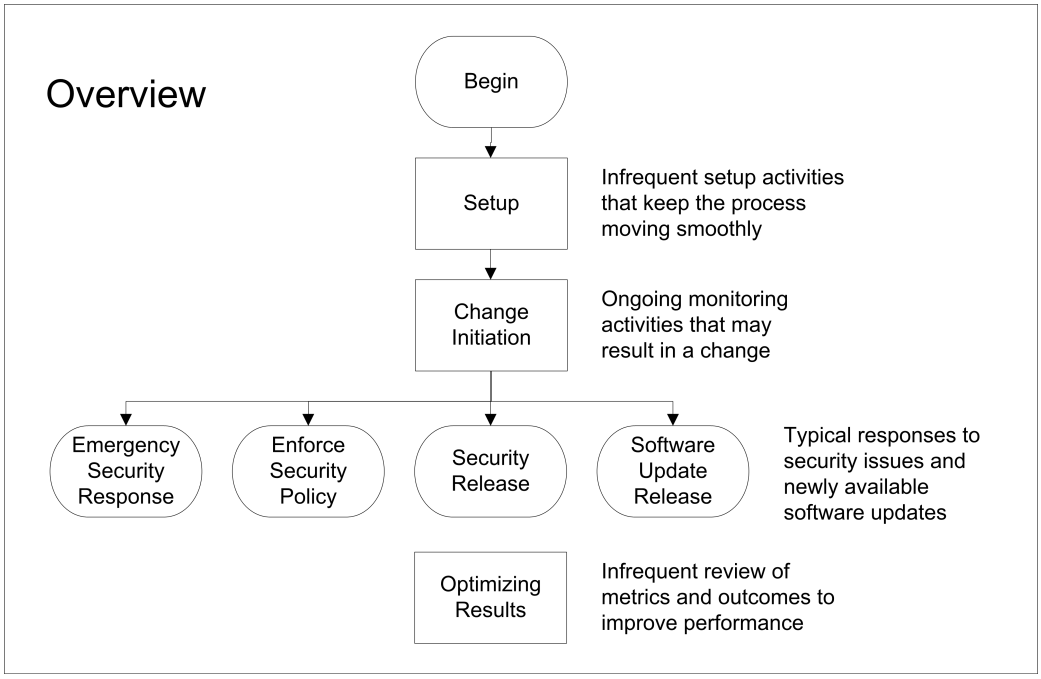


Figure 3.1
Overview of Security Patch Management

Software companies release software updates to resolve problems that become apparent after the release of the product. Keeping an environment secure and reliable to prevent attacks from succeeding requires diligence and effort.

This chapter provides the reader with a high-level understanding of the ongoing security patch management process that every organization should have in place, and describes key concepts and principles of security patch management, including:

- **Setup.** The infrequent activities that are required to support effective security patch management, such as taking inventory and baselining the environment, subscribing to security alerts, establishing security reporting to assist with issue identification, and configuring and maintaining the patch management infrastructure.
- **Change initiation.** Ongoing monitoring that is used to identify any security issues that should be resolved by changing the production environment. This includes reviewing several sources of information and reports to identify new software updates and security issues, determining their relevance, quarantining new software updates for use in subsequent steps, and initiating a response to address the security issue.

The typical responses to an identified security issue covered in this chapter include:

- **Security release.** Releasing a software update or related countermeasures is the usual response to a newly-identified vulnerability. Performing a security release includes change management, release management (including testing), and review (including rollback, if necessary).
- **Enforcing security policy.** This response is necessary when previously addressed vulnerabilities recur in the environment. Recurring vulnerabilities are at increased risk of exploitation by viruses, worms, and attack tools that remotely scan computers for security weaknesses and published vulnerabilities.
- **Emergency security response.** Preparing for and responding to attacks that exploit security vulnerabilities in your organization.

Over time, an organization should focus on optimizing results—infrequent activities that review how effectively an organization performs security patch management and ways of measuring and improving the process to meet specific business objectives. This chapter concludes with a high-level review of optimizing results.

Setup

This section provides a high level overview of the infrequent setup activities for security patch management. This topic is discussed in more detail in Part II, Chapter 2, "Setup."

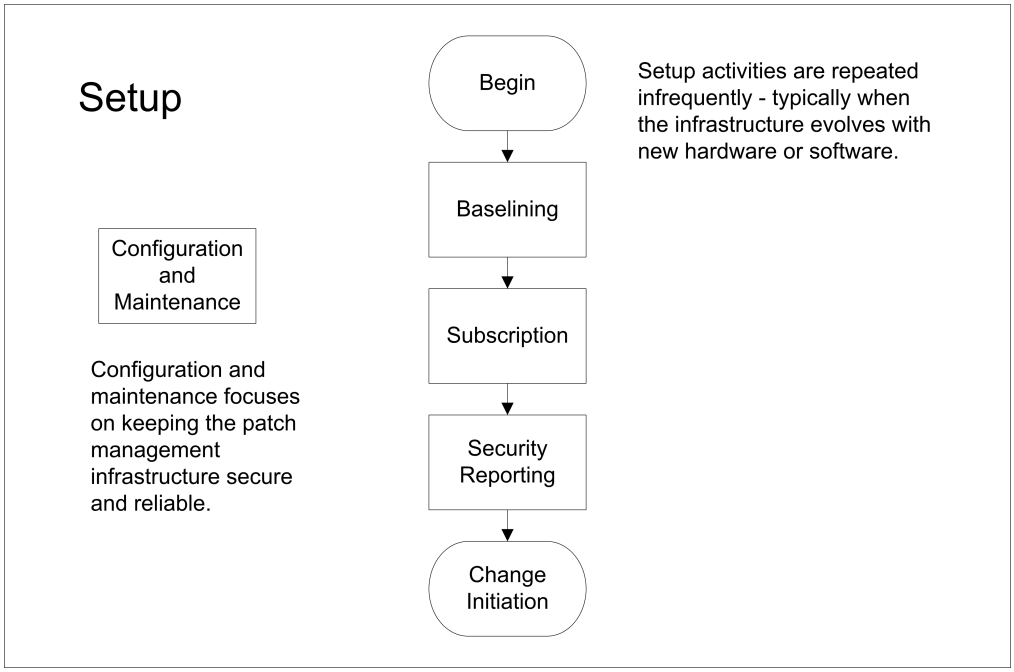


Figure 3.2
Setup for Security Patch Management

Baselining

Baselining is the process of bringing the computers in an environment to a standard software baseline—that is, with the same software versions and software updates. Without baselining, security patch management is unnecessarily complex and difficult to do comprehensively.

Baselining includes the following steps:

1. Generate an inventory of hardware (computers and models), operating systems, and applications. This includes specific software versions and software updates applied.
2. Use the information obtained from the inventory to define standard software baselines for all computers.
3. Perform an audit to determine which computers meet their baseline and which do not—then decide if the baseline is too aggressive and needs to be modified.
4. Take the necessary actions to bring the non-compliant computers up to their required baseline. This involves installing service packs and other software updates or even upgrading software versions.
5. Audit the environment to ensure the standard software baselines are met. There may be a few exceptions necessary for business reasons that can be managed on a case-by-case basis.

Different classes of computers (such as messaging servers and directory servers) may have different application baselines, but may share an operating system baseline. Different models (such as HP and Dell) may have different baselines. Every computer should be categorized and mapped to a defined software baseline including operating system and applications.

In subsequent iterations, baselining will define upgraded software baselines that include new software updates, future service packs, and new software versions.

Note: In an undisciplined environment, baselining will be challenging the first time, as software versions and applied software updates are likely very disorganized. Without baselining however, the results of vulnerability scanning will be overwhelming and almost impossible to manage.

Software baselines can be used when creating image-based installations for new computer deployments and reinstallations. Software baselines are also closely related to, and may become part of, the software standards in your security policy. Both represent a minimum standard software level determined as necessary to keep the environment secure and reliable.

The baselining process has other advantages; aiding security patch management by providing a comprehensive inventory of computers and software that is used in later steps and other operational tasks beyond patch management.

Microsoft Policy on the Product Support Life Cycle

Microsoft only creates security updates for supported products. To ensure access to new security updates, software baselines should only include supported products on a recent service pack. For more information about which versions of Microsoft products are currently supported, see the Microsoft Product Support Lifecycle: http://support.microsoft.com/default.aspx?scid=fh;[LN];lifecycle.

Security patches are released for supported products on the current service pack and the immediately preceding service pack, whenever it is commercially viable.

Asset Categorization and Valuation

When baselining, it is helpful to identify several asset categories and sub-categories that you will use later to prioritize how quickly each is updated and to identify any special testing or deployment considerations, such as services running 24 hours a day, seven days a week with infrequent servicing opportunities.

Keep in mind the cost of downtime and the security needs of each class of asset when creating categories and sub-categories—use risk tolerance to help assign the level of importance to various assets. Categorization by technical function may not always be appropriate—for example, one database server may store less valuable information than another, which stores information that is critical to the survival of the organization.

Note: Regardless of the relative value of each asset, keep in mind that several types of vulnerabilities can be exploited with a denial-of-service attack, which can impact business operations even through assets that are low in value. The goal of security patch management should be as comprehensive as possible—leaving no opening for attack.

Subscription

Subscription is the process of signing up for various information services to ensure notification of software vulnerabilities and related software updates.

The Microsoft Security Response Center (MSRC) responds to all security-related concerns about Microsoft products and prepares security bulletins as a result.

Every organization that uses Microsoft software should subscribe to the Microsoft Security Bulletin Service for notifications of new vulnerabilities and related software updates. When you subscribe to the Microsoft Security Bulletin Service, you can choose to receive:

- **Technical alerts.** To receive technical versions of Microsoft security alerts, subscribe to Product Security Notification: http://www.microsoft.com/technet/security/bulletin/notify.asp.
- **Non-technical alerts.** To receive non-technical versions of Microsoft security alerts, subscribe to Microsoft Security Update: http://register.microsoft.com/subscription/subscribeme.asp?id=166.

Ensure that more than one person in your organization receives security alerts so that you will continue to receive the information that the security bulletins contain even if your IT person is on vacation or otherwise out of the office.

You can search for historical security bulletins at the Microsoft Security Bulletin Service: http://www.microsoft.com/technet/security/current.asp.

Security Reporting

Ongoing scanning and reporting of security issues is crucial for ensuring the security of an environment. This step in the process is where these security reports are established.

The most important security reports that every environment should maintain as input for effective security patch management include:

- **Vulnerability scanning reports.** You can create these reports using tools such as the Microsoft Baseline Security Analyzer (MBSA). May also include scanning for other vulnerabilities, as defined by your security policy.
- **Virus scanning reports.** You can create these reports using the virus tools your organization has selected.
- **Intrusion detection reports.** These reports summarize the results of various activities, including network monitoring and examining event logs and the output of any intrusion detection system in place. An intrusion detection system is software that helps automate the intrusion detection process.

Each of these reports, along with security notifications and other sources of information, keeps the organization informed of security issues and drives the change initiation process.

The generation of these reports should be as automated as possible and they should be generated on a regular basis—daily for most, weekly for some. The frequency depends on the level of automation and scale that can be achieved for each report, the technologies that you have in place, your staffing level, and your organization's level of commitment to a secure environment.

Configuration and Maintenance

The patch management infrastructure encompasses all of the tools and technologies that are used to maintain an inventory, assess software levels, test, deploy, and install software updates, and report on progress and security issues. This infrastructure is a vital instrument for keeping the entire environment secure and reliable. It deserves proper care and attention.

If the patch management infrastructure is vulnerable, it can provide an attacker with significant reach into an organization—providing an opportunity to impact many computers in a short period of time. Your patch management infrastructure must be deployed, configured and maintained with high security considerations.

Change Initiation

This section provides a high level overview of change initiation, the ongoing activities that drive the security patch management process. This topic is discussed in more detail in Part II, Chapter 3, "Change Initiation."

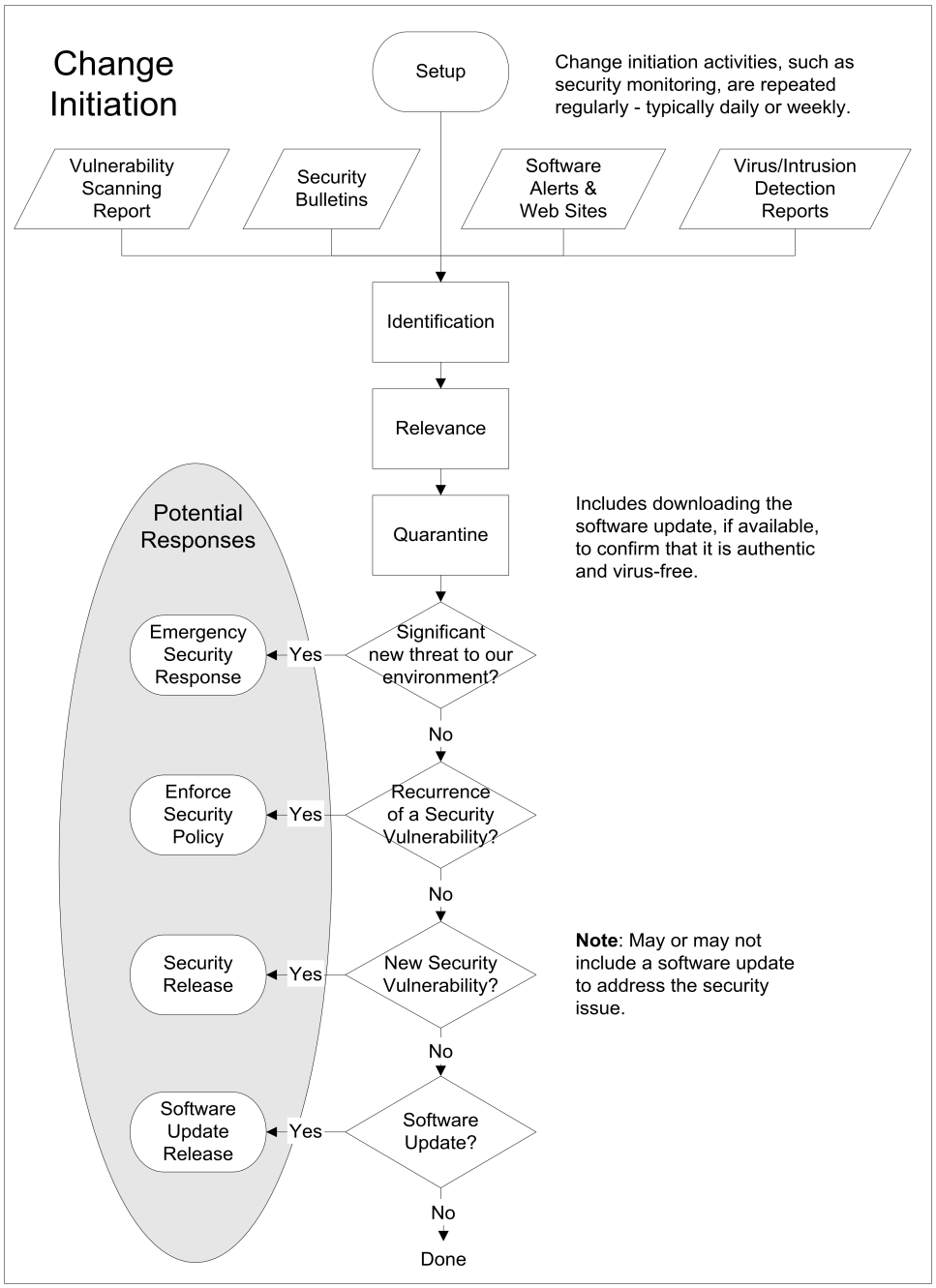


Figure 3.3
Change Initiation for Security Patch Management

Identification

The identification of security issues and related software updates is a regular process that relies on a variety of sources.

Vulnerability Scanning Reports

These reports identify security vulnerabilities that are related to configuration settings and software levels. The objective of security patch management should be to have as few software vulnerabilities as possible.

Vulnerabilities that are related to recently announced issues are typically addressed by initiating a security release. Vulnerabilities recurring in the environment and weak security settings are typically addressed by enforcing security policy.

Security Bulletins

Microsoft security bulletins identify new product vulnerabilities, important updates to past bulletins, and new viruses that have been discovered by others.

Bulletins for vulnerabilities identify related software updates and countermeasures. Security bulletins for relevant new vulnerabilities are typically addressed by initiating a security release.

Microsoft security bulletins are reissued when the severity rating changes. For example, the severity rating can change if an active exploit for a vulnerability becomes widely available. They are also reissued if the referenced software update is re-released; fixing a quality issue that managed to evade the testing Microsoft performs on software updates.

Reissued bulletins that have a higher severity should be evaluated again to determine if an already scheduled security release should be reprioritized and accelerated.

If a newer version of the software update is released, determine if it should be made available to a security release that is already underway, or if another security release with the improved software update should be performed.

Microsoft Policy on Security Bulletins

Microsoft never distributes software directly through e-mail. If you receive a message that claims to contain software from Microsoft, do not run the attachment—delete the message instead.

For more information about these policies, see Microsoft Policies on Software Distribution:
http://www.microsoft.com/technet/security/policy/swdist.asp.

There are several e-mail hoaxes that claim to be from Microsoft. When you receive a Microsoft Security bulletin, confirm it and *all hyperlinks to software updates* by visiting the official Microsoft Security Bulletin Service page:
http://www.microsoft.com/technet/security/current.asp.

For more information about these types of hoaxes, see Information on Bogus Microsoft Security Bulletin E-mails:
http://www.microsoft.com/technet/security/news/patch_hoax.asp.

Software Alerts and Web Sites

Besides reading security bulletins and performing vulnerability scanning, you should regularly visit product and technology Web sites to see if any new service packs, important security considerations, or white papers have been published recently. Some services send e-mail alerts that are related to specific products.

The result of finding new security information could be to initiate a security release, investigate if the security policy should be updated to be more secure, or to investigate if a new tool might be useful in the environment. Relevant software updates that are not related to security should initiate a software update release.

Awareness of several organizations being hit by a new virus, worm, or another attack may be a good reason to proactively initiate an emergency security response. This can help minimize the damage of an attack on the horizon and put your organization is a state of readiness.

Virus and Intrusion Detection Reports

The identification of a virus or an intrusion in the environment indicates an attack in progress that needs to be addressed quickly.

The typical response to an attack is an emergency security response, although you can choose to employ less extreme measures if you know that the attack is well contained.

Microsoft Policy on Newly-Discovered Vulnerabilities

If you identify a new vulnerability within a Microsoft product, please report it to Microsoft immediately. Contact your Technical Account Manager if you have Microsoft Premier Support.

For those organizations that do not have a Microsoft Premier Support agreement, Microsoft provides a Web form in which you can describe the potential vulnerability. Microsoft pursues all potential vulnerabilities. You can Report a Security Vulnerability at: https://www.microsoft.com/technet/security/bulletin/alertus.asp.

Relevance

Each security issue that is identified should be reviewed to determine the relevance of the issue to the environment. Your review should include all associated documentation, such as Knowledge Base (KB) articles. The relevance process should answer the question: Does the vulnerability exist within any specific software versions in our environment? If the vulnerability exists, it needs to be addressed.

Note: A vulnerability that has been mitigated through a countermeasure (such as port filtering or disabling services) is still a source of risk.

Because some countermeasures are not comprehensive, they don't prevent exploits of all kinds. Also, if you change your configurations in the future, you might inadvertently remove countermeasures. For these reasons, even those vulnerabilities that have been mitigated by countermeasures are still considered relevant, and require a security release.

Although the existence of a countermeasure may reduce the priority and resulting schedule of a software update deployment, the issue should still be tracked and eventually addressed by eliminating the underlying vulnerability.

Quarantine

To prevent virus infection or malicious code from affecting your production infrastructure, all software updates should be downloaded and reviewed in an isolated, quarantined environment. This quarantine should be imposed on all software and documentation downloads.

The software update that has been through quarantine should then be the only version used during change and release management.

Note: Microsoft software update distribution tools, including Software Update Services (SUS) and Systems Management Server (SMS) with the SUS Feature Pack, only download software updates from trusted, virus-free, Microsoft sources and confirm the authenticity of software updates by testing digital signatures.

This policy is seen by many as an acceptable mitigation of the risks that are associated with downloading software updates. Other people may still demand a quarantine environment for software updates downloaded by Microsoft software update distribution tools.

Security Release

This section provides a high level overview of a security release, the typical response to a new software vulnerability identified in an environment. This topic is discussed in more detail in Part II, Chapter 4, "Security Release."

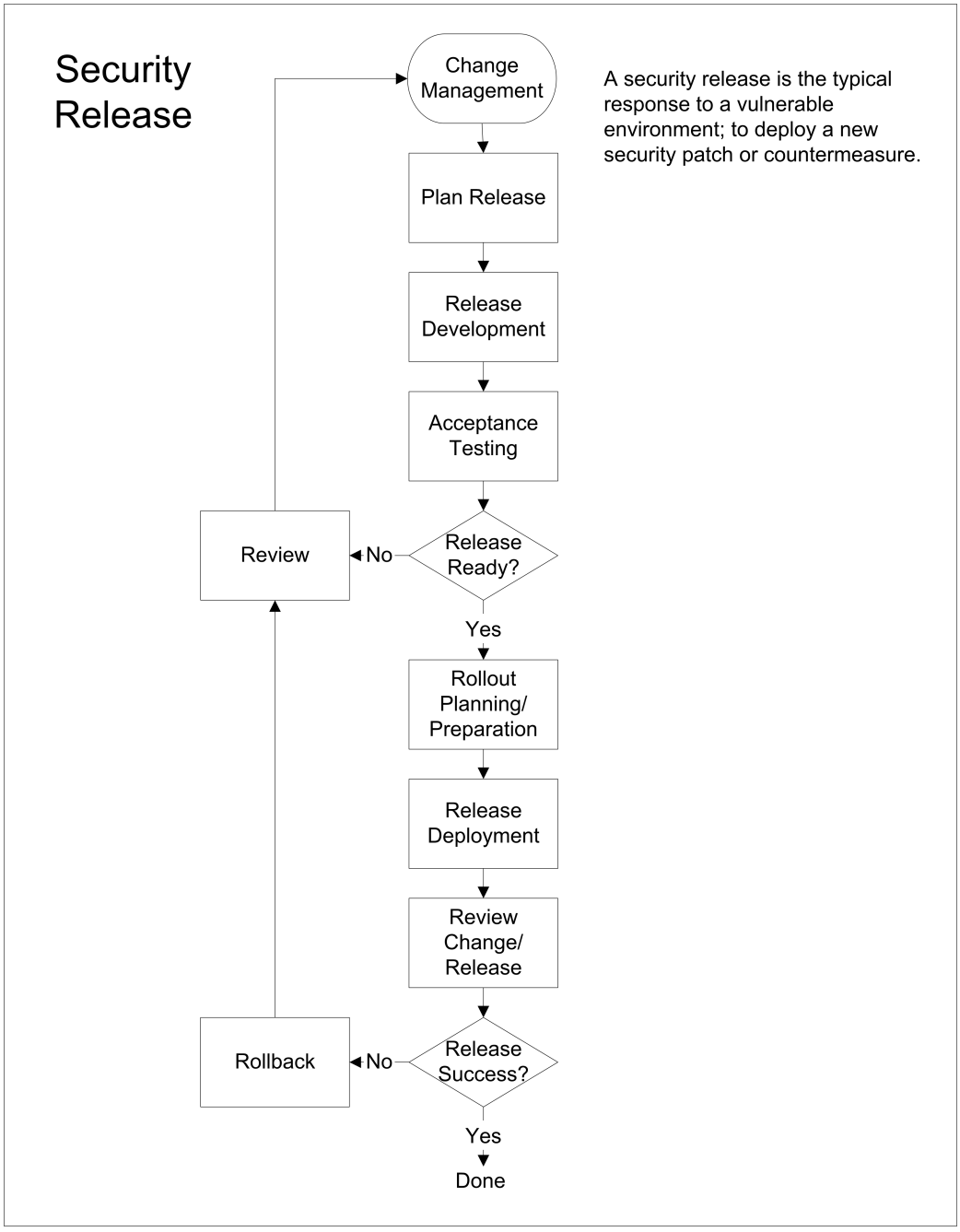


Figure 3.4
Security Release

Change Management

Change management, in the context of a security release, is the process of determining the appropriate response to a software vulnerability or threat, including:

- Determining what kind of change is required in the production environment—deploying a software update, applying countermeasures that mitigate the vulnerability, or both.
- Describing the required change so that others can understand and act on it.
- Prioritizing and scheduling a release to implement the change.
- Ensuring the appropriate people authorize and approve the change and proposed release schedule.

Deciding on Countermeasures

The best response to a software vulnerability is to deploy the software update that fixes it.

Occasionally, this response might not be immediately appropriate. Applying a countermeasure has several potential benefits:

- Some countermeasures can be applied without downtime, whereas some software updates require a restart.
- Some countermeasures are of lower risk and can be applied more quickly and with less aggressive testing, unlike some software updates.
- Countermeasures can usually be undone quickly if they have an unintended impact, whereas some software updates cannot be uninstalled easily.

For these reasons, countermeasures are best used when a speedy response to a vulnerability is required (because of the dangerous nature of a potential exploit), and the software update cannot be applied as quickly.

Whenever you implement a countermeasure-only release, you should also schedule a second security release to install the actual software update and eliminate the underlying vulnerability.

Note: Many countermeasures are good general security practices. If the countermeasures are fully understood, consider making them standard configuration requirements by adding them to the security policy.

Describing the Change

There are several aspects of a security release that you should understand and be able to describe, including:

- What is the change? What vulnerability is the change in response to?
- What services will be impacted by the change?
- Is a software update being deployed?
  - Does the software update require a restart to complete the installation?
  - Can the software update be uninstalled?
- Should any countermeasures be applied?
- What are the recommended test strategies for this change?

Prioritizing and Scheduling the Release

The following table provides some guidelines for various security release priorities.

Table 3.1: Security Release Deployment Timeframe Guidelines

Priority	Recommended Timeframe	Minimum Recommended Timeframe
1	Within 24 hours	Within 2 weeks
2	Within 1 month	Within 2 months
3	Depending on availability, deploy a new service pack or update rollup that includes a fix for this vulnerability within 4 months.	Deploy the software update within 6 months
4	Depending on availability, deploy a new service pack or update rollup that includes a fix for this vulnerability within 1 year.	Deploy the software update within 1 year, or may choose not to deploy at all.

The priority and resulting schedule for a security release should be determined by taking into consideration the defined MSRC severity level of the vulnerability along with aspects that are unique to your environment.

A simple mechanism for determining priority involves mapping the MSRC-defined severity level of the vulnerability to an initial priority level for the release (Table 3.2), then raising or lowering the priority level depending on the needs of your organization and unique aspects of your environment (Table 3.3).

Table 3.2: Determining Initial Priority

MSRC Severity Level of Vulnerability	Initial Priority
Critical	1
Important	2
Moderate	3
Low	4

Table 3.3: Factors That May Influence Release Priority

Environmental/Organizational Factor	Possible Adjustment of Priority
High value or high exposure assets impacted	Raise
Assets historically targeted by attackers	Raise
Mitigating factors in place, such as countermeasures that minimize the threat	Lower
Low value or low exposure assets impacted	Lower

To ease the impact on resources performing release management, multiple security changes for the same category of asset can be combined into a single release. This is most appropriate for priorities 2 through 4, as listed in Table 3.1.

Release Management

The focus of release management is on facilitating the introduction of software and hardware releases into managed IT environments. Typically these managed IT environments include the production environment and managed preproduction environments.

The release management process includes the following steps:

- **Plan release.** Defining and prioritizing all requirements for the release and creating the required release plans including a test plan, rollout plan, and rollback plan. These plans are used by all team members as the release moves through the release management process.
- **Release development.** Selecting the release mechanism and designing, building, and testing the release package. The mechanism should deliver the updates in the most efficient manner and take into account the addition of new computers at points between planned baseline updates.
- **Acceptance testing.** The purpose of testing, to this point, will have been to confirm that the release package works correctly within a development environment. Acceptance testing focuses on how the release and release package performs in an environment that closely mirrors production.
- **Rollout planning and preparation.** Finalizing rollout specifics and preparing the infrastructure for the release. Rollout preparation requires the coordination of resources and may include pre-staging of hardware and software.
- **Release deployment.** Distributing and installing the release across the organization. This may involve a pilot to introduce the patch to a small group of computers initially, and may or may not use a phased deployment approach.

Note: The focus of acceptance testing should be on how a security release will work in your specific environment. Pay particular attention to compatibility with line-of-business (LOB) and independent software vendor (ISV) applications, older or unsupported hardware and software, new hardware, and unique configurations.

You may want to work closely with the software suppliers of your crucial applications to assist you in acceptance testing.

Change Review

The review process is intended to confirm successful deployment of the security release while ensuring there is no unintended negative impact on related operations. This involves checking patch distribution logs and reports, monitoring calls to the help desk, and checking subsequent vulnerability scanning reports to ensure that the vulnerability is addressed.

This is also an opportunity to review how the process worked for this release by having a post-implementation review. Do any details of setup, change initiation, or change and release management need to be adjusted to improve the process the next time?

Despite following proper planning and testing procedures, problems can arise. Even if a particular software update cannot be uninstalled, a rollback approach should have been identified for use during release management in case the security release cannot be fixed through other means.

Enforcing Security Policy

This section provides a high level overview of enforcing security policy, the typical response to vulnerabilities recurring in an environment. This topic is discussed in more detail in Part II, Chapter 5, "Enforcing Security Policy."

New computer installations, lab equipment, mobile users, and decentralized administration can all be sources of previously-addressed vulnerabilities recurring in an environment. Recurring vulnerabilities are at increased risk of exploitation by viruses, worms, and attack tools that remotely scan computers for security weaknesses and published vulnerabilities.

Having a security policy is an obvious prerequisite for enforcing one. A security policy should include definitions of the required software levels, software updates, and security settings that effectively address the risk tolerance level of the organization.

Computers that fail to adhere to the security policy are considered to be vulnerable and a source of unacceptable risk. The vulnerability scanning report created earlier can also be used to identify noncompliance with the security policy so that action can be taken quickly.

Enforcement approaches, tools, and techniques including identifying the owner of a computer, addressing typical vulnerabilities, and escalation strategies for resolution on a timeline are all part of effective security policy enforcement.

Emergency Security Response

This section provides a high level overview of an emergency security response to an attack that is underway or likely to occur in the near future. This topic is discussed in more detail in Part II, Chapter 6, "Emergency Security Response."

There are three main areas that should be prepared *in advance* of an emergency security response:

- Regular auditing and intrusion detection tools and practices.
- An identified incident response team with a detailed incident response plan.
- Established post-incident actions and a review process to learn from the attack.

Auditing and intrusion detection involves monitoring the network and network perimeter, regularly checking the services and configurations of various computers, and the regular monitoring of event logs for specific event IDs across the organization. Several tools are available to assist with these activities, including intrusion detection systems.

The formal incident response plan describes how the incident response team will perform the following:

- **How to evaluate the attack.** This should include areas to investigate and uncover.
- **Who should be notified of the attack.** This list should include law enforcement agencies and legal council.
- **How to isolate and contain the attack.** This should include common contingencies that can be put in place.
- **How to analyze and respond to the attack.** This should include remediation and accelerated change and release management.
- **What should be done after the attack.** This step should include an impact assessment, repeating the release of any changes made to the production environment (to catch issues inadvertently caused by quick changes made under pressure), and a review of the organization's performance during the attack.

Optimizing Results

This section provides a high-level overview of optimizing security patch management results. This topic is discussed in more detail in Part II, Chapter 7, "Optimizing Results."

After performing security patch management for a while, it is important to monitor and improve progress. Even with proper planning, there will be improvements that can be identified over time.

Each element of security patch management should be reviewed, such as setup and change initiation, plus each of the potential responses including security release, enforcing security policy, and emergency security response.

Security patch management should be a standard part of ongoing operations. There are many resources available to help review and improve these operations, including the Microsoft Operations Framework Self-Assessment Tool: http://www.microsoft.com/technet/itsolutions/tandp/opex/moftool.asp.

To learn more about operational assessments and to find consulting services that can perform an operations assessment, see the Operations Assessment Service Offering: http://www.microsoft.com/solutions/msm/evaluation/overview/opsassessment.asp.

4

Tools and Technologies

This chapter highlights the technologies that are available from Microsoft® for security patch management and helps you to make the best choices for your organization.

Executive Summary: Software Update Distribution

There are three basic choices today for software update distribution from Microsoft: Windows® Update (WU), Software Update Services (SUS) 1.0 SP1, and Systems Management Server (SMS) 2.0 with the SUS Feature Pack.

Table 4.1: Executive Summary: WU, SUS, and SMS.

Feature Area	Windows Update	SUS 1.0 SP1	SMS with SUS Feature Pack
Centralized administration	**Poor.** Computers install updates selected by user.	**Good.** Updates approved by administrator.	**Best.** Updates approved by administrator and specifically targeted.
Central inventory	**Poor.** No central inventory or assessment.	**Poor.** No central inventory or assessment.	**Best.** Customizable central inventory for hardware, software, and vulnerabilities.
Software coverage	**Good.** All types of Windows updates. Windows only.	**Fair*.** Security patches, critical updates, updates, and update rollups. Windows only.	**Best.** Distributes any software or software updates to SMS clients.
Cost	**Free.**	**Free.**	**License fees required.**
Windows operating systems	**Good.** Windows 98, Windows 98 SE, Windows Millennium Edition, Windows XP, Windows 2000, and Windows 2003.	**Fair.** Windows XP, Windows 2000, and Windows 2003.	**Best.** Windows 95, Windows 98, Windows 98 SE, Windows Millennium Edition, Windows NT® 4.0, Windows XP, Windows 2000, and Windows 2003.
Reporting	**Poor.** No central reporting.	**Fair.** Some central reporting through log files.	**Best.** Built-in Web reports and customizable reports.
Architecture and install	**Easy.** Client configuration only; no other infrastructure.	**Easy.** Simple server architecture: easy client setup.	**Hard.** Complex architecture and services installation.

* SUS 1.0 SP1 may include service packs in the near future; turning this from Fair to Good.

The previous table provides a simple comparison of each technology across a variety of feature areas that might be of interest to organizations that want to automate security patch management.

Note: Windows Update and Software Update Services 1.0 SP1 only provide software updates for Windows operating systems and Windows components (such as Microsoft Internet Explorer, Microsoft Internet Information Services, and Microsoft Windows Media® player), whereas security vulnerabilities might be identified in any Microsoft product.

As much as 75 percent of all historical Microsoft security bulletins have been for Windows operating systems and Windows components, suggesting that WU and SUS provide fair automation support for most of the security-related software updates.

In WU and SUS environments, several Microsoft products must be updated by using other services such as Office Update or by manually applying software updates (including software updates for Microsoft Exchange and Microsoft SQL Server™).

SMS does not have this limitation and can be used to update any software product on an SMS client.

Software Update Assessment and Reporting

WU and SUS will ascertain if a software update is necessary on each computer, but they do not perform this assessment centrally or provide consolidated vulnerability reports. In WU and SUS environments, Microsoft Baseline Security Analyzer (MBSA) and the Office Update Inventory Tool can be used to centrally assess software updates and vulnerabilities to prepare basic reports.

SMS 2.0 with the SUS Feature Pack includes the capabilities of MBSA and the Office Update Inventory Tool and provides easy-to-use central software update and vulnerability assessment and reporting capabilities.

Microsoft Products and Technologies Roadmap

Microsoft provides small, medium, and large organizations with several tools and services to equip users and administrators with methods for keeping their computers secure and reliable.

Microsoft software update and vulnerability assessment tools:

- **Microsoft Baseline Security Analyzer (MBSA).** MBSA is a security vulnerability analyzer that can assess multiple computers for common security vulnerabilities and missing security-related software updates.
- **Office Update Inventory Tool.** A software update analyzer that can assess multiple computers for missing Office software updates.

Microsoft software update installation services for computers:

- **Windows Update (WU).** Windows Update is an online service that helps keep the Windows operating system and Windows technologies on a computer up to date.
- **Office Update.** Office Update, similar to Windows Update, is a Web site that helps keep the Microsoft Office suite of products on a computer up to date.

Microsoft software update approval and distribution services for organizations:

- **Software Update Services (SUS).** SUS is an extension of the Windows Update service that allows companies to maintain and administer updates for all Windows computers located inside the company firewall.
- **Systems Management Server (SMS) 2.0 and the SUS Feature Pack.** SMS is an enterprise-class change and configuration management solution from Microsoft. It provides full software inventory and management services and sophisticated reporting.

  The SUS feature pack provides easy access to software updates that are available through Windows Update, Office Update, and the Microsoft Download Center while integrating the assessment capabilities of MBSA and the Office Update Inventory Tool to make security patch management easy in organizations of any size.

Note: For a comprehensive list of all Microsoft products and versions scanned and updated using these tools, see the Product Patch Automation Matrix spreadsheet in the Tools and Templates that accompany this guide.

Software Update and Vulnerability Assessment Tools

Microsoft Baseline Security Analyzer (MBSA) 1.1.1

MBSA runs on Windows 2000, Windows XP, and Windows Server 2003 systems and will scan multiple computers for common security vulnerabilities and missing security updates according to the following table.

Table 4.2: MSBA 1.1.1 Scanning Capabilities

Product	Common security vulnerabilities?	Missing security updates?
Windows NT 4.0	Yes	Yes
Windows 2000	Yes	Yes
Windows XP	Yes	Yes
Windows Server 2003	Yes	Yes
Internet Information Services 4.0, 5.0, and 6.0	Yes	Yes
SQL Server 7.0 and SQL Server 2000	Yes	Yes
Internet Explorer 5.01 and later	Yes	Yes
Exchange 5.5 and 2000		Yes
Windows Media Player 6.4 and later		Yes
Office 2000 and Office XP	Yes	*

* See the following note for information about MBSA 1.2 and Office

MBSA provides a graphical interface for viewing reports generated for each computer, and can also be command-line scripted. MBSA copies an XML file stored on the Microsoft Download Center to ensure a current list of assessment details for new security-related software updates.

Note: MBSA 1.2 will include the Office update scanning capabilities of the Office Update Inventory Tool.

For More Information

Microsoft Baseline Security Analyzer:
http://www.microsoft.com/technet/security/tools/tools/mbsahome.asp.

Office Update Inventory Tool

The Office Update Inventory Tool enables administrators to check one or more computers for the status of Microsoft Office 2000 and Office XP updates. Administrators can run the tool from a central location to check the status of multiple computers.

The tool produces a report used to determine which updates have been applied, which updates are available to be applied, and which updates can be applied only to an administrative image.

For More Information

Office Update Inventory Tool:
http://www.microsoft.com/office/ork/xp/journ/OffUTool.htm.

Software Update Installation Services for Computers

Windows Update

Windows Update is a free service to keep Windows computers up to date with the latest software updates. Windows Update is made up of three components: the Windows Update Web site, the Automatic Update client, and the Windows Update Catalog.

Windows Update Web Site

Millions of people use the Windows Update Web site each week as a way to keep their Windows systems current. When users connect to the Windows Update site, their computer is evaluated to check which software updates and updated device drivers (in the "Designed for Windows" Logo program) should be applied to keep their system secure and reliable.

Windows Update can automatically notify users of critical updates and security patches through the Automatic Updates client—without visiting the Windows Update Web site.

Automatic Updates Client

Available in Windows 2000 SP3, Windows XP Home Edition and Windows XP Professional, the Automatic Updates client provides notification services for Windows Update and provides a user interface for configuring automated download and installation preferences.

Note: The Critical Update Notification (CUN) client introduced in the Security Toolkit is now obsolete. If you have not done so already, update to Windows 2000 SP3 or later to use the latest Automatic Updates client.

Windows Update Catalog

The Windows Update Web site includes a catalog of all software update installation packages for downloading by administrators.

These software update installation packages can then be stored on CD, distributed, and installed through other means, such as SMS or third-party software distribution tools, or used when installing new computers.

For More Information

Windows Update Frequently Asked Questions:
http://support.microsoft.com/support/windows/update/faq.

Windows Update Catalog:
http://windowsupdate.microsoft.com/catalog.

Designed for Windows Logo program:
http://www.microsoft.com/winlogo.

Office Update

The Microsoft Office Product Updates Web site is designed to determine which Office suite is installed on your computer, and then offer missing software updates for all applications that are included in the suite. Office Update is available for versions of Office 2000 and later. Supported Office products include Word, Outlook, PowerPoint, Access, FrontPage, Publisher, InfoPath, OneNote, Visio, and Microsoft Project.

Office Update functions similar to the way the Windows Update operates, except that there is no automatic notification for Office as is provided for Windows by the Automatic Updates client.

Office Download Center

Like the Windows Update Catalog, the Office Download Center provides a comprehensive catalog of software update packages that can be downloaded, stored, distributed, and installed as you like.

For More Information

About Office Product Updates:
http://office.microsoft.com/productupdates/aboutproductupdates.aspx.

Office Download Center:
http://office.microsoft.com/downloads.

Software Update Distribution Services for Organizations

Software Update Services 1.0 SP1

Software Update Services (SUS) 1.0 SP1 is a version of Windows Update designed for organizations that want to approve each software update before it is installed in their environment.

SUS allows administrators to very quickly and easily deploy Windows-related security patches, critical updates, updates, and update rollups to any computers running Windows 2000, Windows XP Professional, or Windows Server 2003.

SUS includes the following capabilities:

- Software updates can be approved uniquely on each SUS server; enabling testing in a separate environment as well as phased deployments across the enterprise.
- Software updates can be distributed through SUS (saving bandwidth on shared Internet connections), or SUS clients can be configured to download software updates from Windows Update.
- SUS can provide many Windows Update software updates to computers that don't have Internet access.
- The SUS server architecture is made up of simple parent-child relationships and can scale to very large environments—each SUS server can support up to 15,000 clients.
- Software updates can be copied by CD from a SUS server that is connected to the Internet to a SUS server architecture with no Internet access.

SUS servers require Windows 2000 Server or Windows Server 2003, Internet Information Services, and port 80 for communications with SUS clients. Every SUS server can be configured to synchronize software update packages and approvals either manually or automatically from its parent SUS server, enabling flexibility in how the environment is maintained.

SUS clients use the exact same Automatic Updates client as is used by Windows Update. Clients are configured to connect to specific servers, and can be configured for automatic software update installations or end-user prompting.

Note: SUS only provides security patches, critical updates, updates, and update rollups available from Windows Update. Service packs may be available through SUS in the near future. Device driver updates are not provided through SUS.

For More Information

Software Update Services FAQ:
http://www.microsoft.com/windows2000/windowsupdate/sus/susfaq.asp.

Software Update Services Deployment white paper:
http://www.microsoft.com/windows2000/windowsupdate/sus/susdeployment.asp.

Microsoft Solutions for Management (MSM) Patch Management Guides for Software Update Services (available August 2003):

- Architecture Guide: http://go.microsoft.com/fwlink/?LinkId=17690.
- Test Guide: http://go.microsoft.com/fwlink/?LinkId=17693.
- Test Case Detail: http://go.microsoft.com/fwlink/?LinkId=17694.
- Deployment Guide: http://go.microsoft.com/fwlink/?LinkId=17691.
- Operations Guide: http://go.microsoft.com/fwlink/?LinkId=17692.

Systems Management Server (SMS) 2.0 and the SUS Feature Pack

With Systems Management Server (SMS) 2.0 and the SUS Feature Pack, administrators are able to easily manage security updates throughout the enterprise. SMS has always been able to distribute any type of software, but the SUS Feature Pack adds functionality that streamlines the security patch management process.

SMS includes comprehensive inventory, vulnerability, and software update assessment capabilities, Web-based reports to show compliance and installation results, and wizards that simplify security patch management.

The SUS Feature Pack for SMS 2.0 is designed to quickly and effectively assess and deploy security patches for Windows, Office, and other products scanned by MBSA. The Feature Pack provides the following new tools for SMS:

- Security Update Inventory Tool
- Microsoft Office Inventory Tool for Updates
- Distribute Software Updates Wizard
- Web Reports Add-in for Software Updates

Security Update Inventory Tool

The Security Update Inventory Tool builds upon SMS inventory capabilities and takes advantage of the power of MBSA to scan each client for security updates. The resulting data is included in the SMS inventory, and a comprehensive status is provided through Web-based reports.

Microsoft Office Inventory Tool for Updates

The Microsoft Office Inventory Tool for Updates uses the existing Microsoft Office Inventory Tool to carry out automated, ongoing scans of SMS clients for installed or applicable Office updates. This data is converted and included in the SMS inventory, and can also be viewed through Web-based reports.

Distribute Software Updates Wizard

The Distribute Software Updates Wizard compares available updates with the inventory of client computers to determine missing and previously-installed updates. Only the necessary updates are installed, whereas redundant or unnecessary updates are ignored or postponed, reducing system overhead.

The Distribute Software Updates Wizard provides the following capabilities:

- Update status for all clients added to the inventory, based on new security update information.
- Review and authorization of updates identified as missing.
- Packages and advertisements built tailored to each update or set of updates.
- Update advertisements distributed to computers using SMS software distribution capabilities.
- Windows Update–style notifications and a rich end-user experience, such as not forcing applications to close when they have not been saved.
- Timers that allow users to save and close applications, and optionally enabling users to postpone updates or to choose not to restart their system.

Web Reports Add-in for Software Updates

The Web Reports Add-in for Software Updates provides a patch management reporting solution, allowing inventory information and reports to be viewed from a Web browser.

The preconfigured reports available from the Web Reports Add-in help track software update status for:

- Individual updates or groups of updates.
- Individual computers or groups of computers.
- All updates or all computers in the enterprise.
- Patches by operating system.
- Detection rate for specific updates.
- Applicable updates by type.

Custom inventory reports and collections can also be created.

For More Information

SMS Software Update Services Feature Pack Overview:
http://www.microsoft.com/smserver/evaluation/overview/featurepacks/fpfaq.asp.

SMS Software Update Services Deployment Guide (available by August 2003):
http://go.microsoft.com/fwlink/?LinkId=17452.

Systems Management Server Web site:
http://www.microsoft.com/smserver.

How the SMS Software Update Services Feature Pack Works:
http://www.microsoft.com/smserver/techinfo/administration/20/using/suspackhowto.asp.

Software Update Management Using SMS 2.0 Software Update Services Feature Pack:
http://www.microsoft.com/technet/prodtechnol/sms/deploy/confeat/smsfpdep.asp.

Microsoft Solutions for Management (MSM) Patch Management Guides for Systems Management Server (SMS) (available by August 2003):

- Architecture Guide: http://go.microsoft.com/fwlink/?LinkId=17684.
- Test Guide: http://go.microsoft.com/fwlink/?LinkId=17687.
- Test Case Detail: http://go.microsoft.com/fwlink/?LinkId=17688.
- Deployment Guide: http://go.microsoft.com/fwlink/?LinkId=17686.
- Operations Guide: http://go.microsoft.com/fwlink/?LinkId=17689.

Appendix A

Third-Party Tools and Resources

This appendix lists a few third-party tools that are related to security patch management.

Several tool categories are not repeated in this appendix, including anti-virus protection, database security, encryption, Internet and network security (including firewalls), enterprise infrastructure tools, backup, and network and enterprise management. These categories are provided online in the Windows Catalog for Software: http://www.microsoft.com/windows/catalog/default.aspx?subid=22&xslt=software.

The following categories are not well-represented in the online catalog, so they are included in this appendix:

- Patch management.
- Intrusion detection systems.
- Forensic investigation.
- Port scanning and enumeration tools.

Note: The following product lists are not comprehensive and are not endorsed by Microsoft.

Patch Management

Table A.1: Third-Party Patch Management Products

Product	Company	Web Site
BigFix Patch Manager	BigFix	www.bigfix.com
by-Control for Windows	BindView	www.bindview.com
Ecora PatchLite and Ecora Patch Manager	Ecora	www.ecora.com
Service Pack Manager	Gravity Storm Software	www.securitybastion.com
RealSecure Vulnerability Assessment	Internet Security Systems	www.iss.net
PatchLink Update 4.0	PatchLink	www.patchlink.com
HFNetChkLt and HTNetChkPro	Shavlik	www.shavlik.com
UpdateEXPERT	St. Bernard Software	www.stbernard.com

Note: Many enterprise management and software distribution tools can accommodate the deployment of software updates. These tools are listed in the Windows Catalog for Software and are not repeated here.

Security Software

The following table lists several products that are useful for intrusion detection, forensic investigation, port scanning, and enumeration.

Table A.2: Third-Party Security Products

Product	Company	Web Site
NMap for Windows		www.nmapwin.org
Captus IPS	Captus Networks	www.captusnetworks.com
PureSecure	Demarc Security	www.demarc.com
Superscan and Scanline	Foundstone	www.foundstone.com
LANguard Network Scanner	GFI	www.gfi.com
EnCase Enterprise and EnCase Forensic	Guidance Software	www.guideancesoftware.com
RealSecure Intrusion Protection	Internet Security Systems	www.iss.net

Part II

The Security Patch Management Life Cycle

1

Introduction

Overview of the Security Patch Management Life Cycle

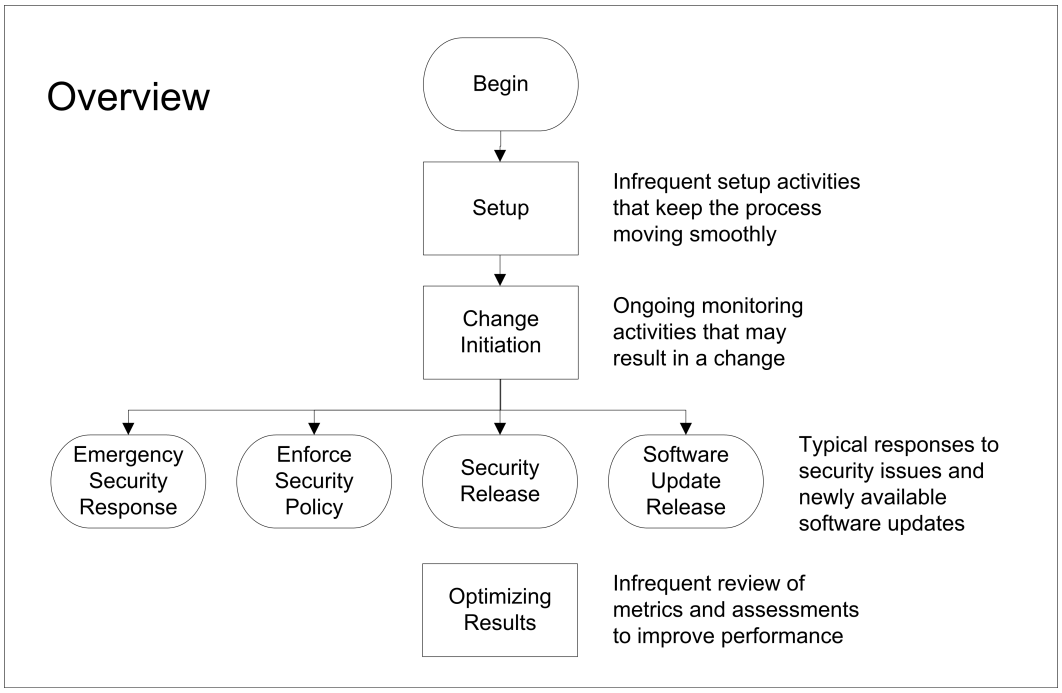


Figure 1.1
Overview of the Security Patch Management Life Cycle

The security patch management life cycle is made up of several infrequent, ongoing, and as-needed activities that are repeated as part of normal operations.

Each of these activities are covered in more detail throughout this Part of this guide, providing process information, techniques, tools, and templates that can help organizations establish effective security patch management.

Note: The Security Patch Management Tasklist included with the Tools and Templates that accompany this guide provides a quick list of tasks that occur throughout the security patch management life cycle. Additionally, all of the process diagrams within this guide are provided in the included Sample Security Processes diagram.

Infrequent Activities

Setup

Setup activities are required to support smooth and efficient security patch management. Setup includes:

- Taking inventory and baselining the environment.
- Subscribing to security alerts and other information sources.
- Establishing security reporting to assist with issue identification.
- Configuring and maintaining the patch management infrastructure.

Setup is typically performed as your environment changes with the introduction of new hardware or software. (Maintenance is performed more frequently to ensure the patch management infrastructure stays secure and reliable.)

Setup activities are covered in Part II, Chapter 2, "Setup."

Optimizing Results

Optimizing results is the infrequent process of examining the metrics and outcomes of security patch management through a security or operational assessment. This process may be done just for security patch management, or patch management could be included as part of a more comprehensive assessment.

Ideally, the result of a security patch management assessment includes:

- A summary of security patch management performance to date.
- A prioritized list of recommended actions that will positively impact future results.

Optimizing results is typically performed quarterly (quick assessment) and annually (rigorous assessment), or as performance improvements are required.

Optimizing results is covered in Part II, Chapter 7, "Optimizing Results."

Ongoing Activities

Change Initiation

Change initiation is typically performed on a daily or weekly basis, and includes the following activities:

- Regularly reviewing Web sites, security notifications, and security reports to identify new software updates and security issues.
- Determining the relevance of the updates and issues in the environment.
- Downloading and quarantining new software updates for use in subsequent steps.
- Initiating a response that appropriately addresses the security issue.

Change Initiation is covered in Part II, Chapter 3, "Change Initiation."

As Needed Activities

The following activities are the typical responses to identified security issues or new software updates that are related to security patch management.

Security Release

Releasing a security patch or related countermeasure is the most frequent response to the identification of a new security vulnerability in an environment. A security release follows the basic release process, which includes:

- Change management. The process of understanding the issue, categorizing and prioritizing the change, and getting approval on making a change to the production environment.
- Release management. The process of planning, developing, testing, and deploying a change in the production environment.
- Change review. This step can include rollback, if necessary due to negative business impact or other quality reasons.

Performing a security release is covered in Part II, Chapter 4, "Security Release."

Enforce Security Policy

Very few IT environments are 100 percent centrally administered through a disciplined process that ensures continuous compliance with evolving security policy.

Every time unmanaged changes to a production environment are made, the opportunity for vulnerabilities to be introduced or reintroduced to an environment exists.

New computer installations, lab equipment, mobile users, administrative users, decentralized or federated administration, and undisciplined change and release management can all be sources of previously addressed vulnerabilities recurring in an environment.

Recurring vulnerabilities are at increased risk of exploitation by viruses, worms, and attack tools that remotely scan computers for security weaknesses and published vulnerabilities.

During security patch management, regular vulnerability scanning reports should identify any vulnerability that violates an organization's security policy.

Vulnerabilities recurring in the environment should be handled by the service desk using a standard incident response process, with an associated escalation strategy and timelines if the vulnerability cannot be addressed appropriately.

Enforcing security policy is covered in Part II, Chapter 5, "Enforcing Security Policy."

Emergency Security Response

This chapter describes how to prepare for an emergency caused by exploited security vulnerabilities and the critical information, steps, and best practices that are necessary to respond effectively if your organization's information technology is at risk or under attack.

Emergency security response is covered in Part II, Chapter 6, "Emergency Security Response."

Software Update Release

Non-security related software update releases are not covered specifically in this guide.

However, deploying a non-security software update can be performed in a similar manner as a security release, which is covered in Part II, Chapter 4, "Security Release."

Security Patch Management and Risk Management

Security patch management is the process of applying software updates and related countermeasures to mitigate the risk of future attacks exploiting security vulnerabilities in your environment. By its very definition, security patch management is a risk management activity.

Accordingly, risk management techniques can be applied to many of the activities of security patch management. The following are a few examples:

- The priority and resulting timeline for a security patch release can be determined in a similar manner to the way a risk is prioritized and mitigated.
- Escalation strategies and timelines when enforcing security policy is a risk mitigation strategy.
- Proactively preparing for an emergency security response is an example of contingency planning.
- Selecting a test strategy for a software update is a way of mitigating the risk that a software update negatively impacts your environment.

This guide does not provide detail on the generic process of risk management, but rather focuses purely on the necessary elements of security patch management. However, it is useful to understand the relationship between risk management and security patch management—your organization may want to become more disciplined at using risk management as a part of your overall security strategy.

Note: For more information on risk management, see the following resources.

Understanding the Security Risk Management Discipline:
http://www.microsoft.com/technet/security/prodtech/windows/secwin2k/03secrsk.asp

Microsoft® Operations Framework Risk Model for Operations:
http://www.microsoft.com/technet/itsolutions/tandp/opex/mofrl/mofrisk.asp

---

Introduction to Techniques

In Part II of this guide, Chapter 2, "Setup" and Chapter 4, "Security Release" both include a Techniques section at the end of the chapter.

Techniques provide prescriptive technical guidance and pull together several resources to help support a technical activity. Techniques often reference the use of a software tool or technology, or one of the Tools and Templates that accompany the guide.

Techniques are provided at the end of each of these two chapters to enable readers to understand the process first, and then perform the process using the appropriate tools by following a technique. In cases in which tools provide overlapping capabilities, techniques can have similar purposes but use different tools.

If you use third-party tools for patch assessment or patch distribution, you may still find the techniques useful in your environment. Regardless, the process and policy details of each chapter are practical—they are generally the same regardless of the tools that you use.

Each Techniques section begins with a table that summarizes the techniques that are available in that chapter, and indicates which Microsoft patch distribution infrastructure the technique is appropriate for: Microsoft Windows® Update (WU), Software Update Services (SUS), or Systems Management Server (SMS). Techniques may be appropriate for more than one of these environments.

Throughout the techniques section of the chapter, each technique is introduced with the following information:

Environment:	The Microsoft patch distribution infrastructure(s) for which the technique is appropriate (WU, SUS, or SMS).
Purpose:	A description of the task that the technique helps accomplish.
Prerequisites:	A list of the tools and capabilities that are required before the technique can be used.
Scale:	An indication of the level of scale the technique is appropriate for, and how it might be scaled differently.

The techniques provided in this guide are not comprehensive. Some organizations will have more or less work to do depending on the management solutions that are already in place.

For example, it is assumed that larger organizations are more likely to have a solution in place to manage their computer inventory. The WU and SUS techniques that are described in this guide can help capture a basic inventory in a small-to-medium–sized environment, but have not been scaled up to replace the functionality that is provided by an enterprise change and configuration management solution such as SMS.

Any IT organization is well-served by having a change and configuration management solution in place to support security patch management and dozens of other operational activities.

Note: The techniques provided in this guide are intended to be prescriptive examples. Each technique and the tools within it should be fully understood and tested for impact before you attempt to use them in your production environment.

2

Setup

Summary

The following sections describe the critical setup activities that are required for the successful implementation of patch management in an organization. These activities include the initial configuration of clients, creating system baselines, subscribing to security alert mechanisms, and implementing vulnerability scanning, detection, and reporting methods.

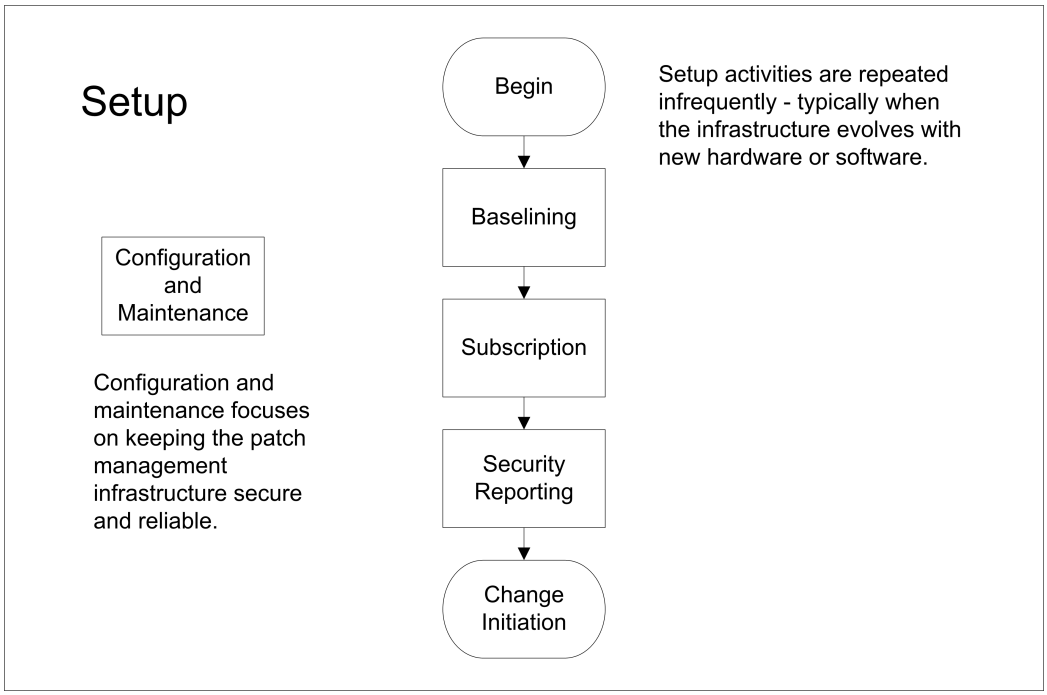


Figure 3.1
Setup Activities for Security Patch Management

Techniques for accomplishing these setup activities are outlined at the end of this chapter.

Infrastructure Configuration and Maintenance

All organizations, regardless of their size, should use automated tools that make administrators aware of available updates, and that provide some control over the installation of security patches.

You must first ensure that your patch management infrastructure is properly configured and documented. With the infrastructure in place, you will need to implement the necessary provisions so that configured clients can use the infrastructure correctly.

Note: You can efficiently customize the configuration settings of the Automatic Updates (AU) client for computers in your environment by using a variety of methods. Customizing the configuration settings allows you to control how your clients interact with some of the patch management technologies that are available from Microsoft®. For more information about some of the available methods to configure clients, see: How to Configure Automatic Updates by Using Group Policy or Registry Settings: http://support.microsoft.com/?kbid=328010.

Additionally, the techniques section at the end of this chapter provides a simple method for configuring AU clients remotely.

Baselining

A baseline is the configuration of a product or system that is established at a specific point in time. An application or software baseline, for example, provides the ability to rebuild a computer to a specific state. It might be necessary to establish baselines for different software applications, hardware vendors, or types of computer. For example, a baseline defined for HP servers running Microsoft Windows® 2000 might differ from a baseline defined for Dell servers.

Baselining first requires that you perform and maintain an accurate inventory of the computers and services within your environment. Often, patches are applied inconsistently throughout an organization, and there is little or no documentation on the reasons for deploying a patch and the computers targeted by a patch deployment. To create the necessary baselines you must, at a minimum, know the following about your environment:

- Operating systems. Which operating systems and which versions are present in your environment?
- Software. Which specific software programs and related versions are in use in your environment?
- Patches. Which service pack versions, software updates, and configuration changes, such as registry modifications, are present in your environment?
- Contact information. How can you contact the individuals or groups who are responsible for maintaining each system in your environment?
- Countermeasures. Which countermeasures to address security vulnerabilities have been deployed in your environment?
- Assets. Which hardware and software assets exist in your environment, and what is their relative value?

Note: It is highly recommended that you make this inventory information available to all those who are involved in your patch management process and ensure that the information is kept up to date.

Operational baselines include all software that is required by different types of computers to operate safely in your production environment. They include operating system version and application versions plus all required software updates. A number of baselines might be required, depending on the different types of hardware and software that are deployed in your environment.

Baseline categories can be different, depending on a variety of factors, including the number of computers included in each category, common software used, and the cost of upgrading a baseline to a certain level. Define each new category when the software components deviate significantly from another baseline.

Baselines should be used during initial computer installations.

After you have compiled the inventory, define baselines that reflect the needs of your production environment.

Note: To assist you with reducing the amount of time that is required to securely introduce new computers into your environment, ensure that your build baselines are incorporated into your automated computer deployments processes (for example, unattended installs, Remote Installation Services [RIS], or image-based installations).

As a best practice, it is recommended that you create a definitive software library to store computer images and all software that is deployed in your environment in a central location where it can be easily accessed for contingency planning and installation purposes.

This chapter includes techniques that can be used to help perform baselining in Microsoft Windows Update (WU), Microsoft Software Updates Services (SUS), and Microsoft Systems Management Server (SMS) environments. See the techniques section at the end of this chapter for more information.

Microsoft Policy on the Product Support Life Cycle

Microsoft only creates security updates for supported products. To ensure access to new security updates, software baselines should only include supported products on a recent service pack. For more information about which versions of Microsoft products are currently supported, see the Microsoft Product Support Lifecycle: http://support.microsoft.com/default.aspx?scid=fh;[LN];lifecycle.

Security patches are released for supported products on the current service pack and the immediately preceding service pack, whenever it is commercially viable.

Update your build baselines to include service packs whenever possible. Doing so greatly reduces the number of patches new computers added to your environment will require to achieve the most secure state possible. For more information on the advantages of using service packs, see Why Service Packs are Better Than Patches: http://www.microsoft.com/technet/columns/security/essays/srvpatch.asp.

Asset Categorization and Valuation

To assist you with determining which updates are relevant to your environment, as well as the order in which they should be applied, you can create categories for the computers in your environment based on their function and how critical they are to proper business operation.

With the appropriate categories in place, you can more efficiently inventory and monitor your environment as security patches are released. Every security bulletin needs to be carefully evaluated to assess the severity that the vulnerabilities pose to your environment. However, not every update needs to be deployed to every computer in your environment. When you determine that an update must be deployed to your environment, having your assets properly categorized will facilitate a smooth and quick deployment.

Computer Categories Help Define Vulnerability Scope

The following table lists some common computer categories based on the roles and services that are provided by computers that might exist in your environment.

Table 2.1: Common Computer Categories and Roles

Category	Role
Directory services	Domain controllers
Infrastructure servers	DNS servers WINS servers DHCP servers File servers Print servers
Messaging servers	Exchange servers
Database servers	SQL Servers
Web servers	Internet IIS servers Intranet IIS servers
Application servers	Terminal Services application servers
Software Distribution Servers	RIS servers SMS servers
End-user client computers	Laptops Desktops Tablet PCs Common area\Kiosk desktops

Regardless of the number of categories that exist in your environment, asset categorization can be helpful when developing proper testing strategies as well as with establishing appropriate deployment timelines. Environments that have few categories can use asset categorization to test patches for deployment by identifying computers that can always be patched first.

For example, your environment might contain a group of common area desktop computers that are used only for access to your intranet. You might decide always to target this category of computers during the initial deployment of a security patch.

Conversely, environments that have many categories can use asset categorization to determine the appropriate deployment timeline for a security patch that will be deployed throughout your environment. For example, desktop computers might get updated any day after regular business hours, whereas messaging servers can only be updated on Saturdays between 5 P.M. and 10 P.M.

Asset Valuation Helps Prioritize Patch Deployment

Determining which computers are most critical to your environment can also be accomplished if your computers are properly categorized. Use the categories to determine which computers are updated first when a security bulletin describing a vulnerability that could have widespread impact to your environment is released. For example, you might want to address a Microsoft Internet Information Services (IIS) vulnerability on your Internet IIS servers before addressing the same vulnerability on your intranet IIS servers because the Internet servers have increased exposure and they are of greater importance to the proper operation of your business.

Furthermore, the service availability requirements in place for the different types of computers can vary greatly within your environment, and could dictate when an update can be deployed to certain computers. For example, restarting Microsoft Exchange servers might be limited to maintenance windows during the weekend, so any updates that require a computer to be restarted could only be deployed during the specified maintenance window.

Unmanaged Computers

Identifying the security patch needs of stand-alone computers or computers that are not members of a domain under your control can be very challenging. Without local administrative rights to these unmanaged clients, collecting system information beyond computer name and IP address can prove very difficult. However, this basic information can serve as the basis for identifying unmanaged clients in your environment.

Microsoft Baseline Security Analyzer (MBSA) scans will report the names and IP addresses of computers that it cannot scan when the user account that initiates the scan does not have administrative rights to the target computers. You can then use this information in conjunction with a port-scanning utility to determine what services the unmanaged computers provide.

For example, a computer listening on TCP port 1433 is likely running Microsoft SQL Server™, which is significant because a computer running SQL Server is likely to be susceptible to SQL Server vulnerabilities. You should take action to resolve the potential vulnerabilities.

Note: For a list of sample port-scanning utilities, see Part I, Appendix A, "Third-Party Tools and Resources".

For information on common port assignments, see Port Assignments for Commonly-Used Services: http://www.microsoft.com/windows2000/techinfo/reskit/samplechapters/cnfc/cnfc_por_simw.asp.

Remember that computers that are secured with IPSec filters or firewall software can cause port-scanning utilities to report incomplete results. For information on configuring IPSec filters to allow for scanning from trusted sources, see the Windows Server 2003 Security Guide: http://go.microsoft.com/fwlink/?LinkId=14845.

In many environments, it might be necessary to implement policies and procedures for handling unmanaged computers. These policies and procedures could require that administrators take action to ensure that computers are updated properly, and what network operations personnel should do if they determine that the threat posed by an unmanaged computer endangers the rest of the network. A common practice is to require that administrators of unmanaged computers regularly schedule local MBSA scans that post their reports to a common share or internal FTP site. Analyzing these results will help determine which computers require security patches. Furthermore, network operations personnel can be authorized to remove computers from the network if the state of their security patches cannot be assessed.

However, such policies and procedures are often difficult to enforce. Performing manual checks of unmanaged computers with the assistance of their respective administrators is the only way to ensure that an unmanaged computer has the required security patches installed.

Subscription

Subscription is the process of identifying the communication mechanisms that pertain to technologies in your environment so any relevant patch and vulnerability notifications can be received as soon as they are made available. Subscribing to the proper notification methods is essential to maintaining and updating your established operating baselines and for implementing an efficient patch management process.

The Microsoft Security Response Center (MSRC) investigates issues that are reported directly to Microsoft, as well as issues discussed in certain popular security newsgroups. The security bulletins that Microsoft releases include summary information describing vulnerabilities and the products that they affect. The bulletins also include detailed technical information describing vulnerabilities and updates and workarounds, as well as deployment considerations and download instructions for any available updates.

Security bulletins and related security patches are released on Wednesdays between 10 A.M. and 11 A.M. PST unless Microsoft determines that customers will be better served by releasing a security bulletin at a different time. This policy was established in response to international customer feedback, with the purpose of better enabling customers to proactively schedule patch management plans.

You can review all security bulletins and other information about Microsoft product security at http://www.microsoft.com/technet/security. All security patches included in the last two service packs for all currently-supported products are available for download from this location.

E-mail Alerts for Technical Users and Medium to Large Businesses

For customers who have more extensive knowledge of or interest in the technology behind security updates, Microsoft TechNet offers the Microsoft Security Notification Service, a free e-mail notification service. These e-mail messages are geared toward IT professionals and contain in-depth technical information. For more information or to sign up to receive the Microsoft Security Notification Service, see: http://www.microsoft.com/technet/security/bulletin/notify.asp.

E-mail Alerts for Home Users and Small Businesses

Microsoft now offers Microsoft Security Update, a free e-mail alert service that makes it easier for small businesses to stay apprised of the latest security updates. Each time Microsoft releases an update, subscribers receive an e-mail message that explains in non-technical terms why Microsoft has issued the update, lists which products are affected, and provides a link to the full announcement on the Security and Privacy Web site. You can sign up to receive the Microsoft Security Update at: http://register.microsoft.com/subscription/subscribeme.asp?id=166

Note: Signing up for the Microsoft Security Update e-mail notification service requires a Microsoft Passport.

Virus Alerts

Microsoft recently joined Network Associates and Trend Micro in the Virus Information Alliance (VIA) in an effort to provide authoritative and timely information to customers on newly discovered viruses. Information on this alliance is available at: http://www.microsoft.com/technet/security/virus/via.asp

Customers can find virus alerts issued by the Product Support Services (PSS) Security Response Team at: http://www.microsoft.com/technet/security/virus/alerts/default.asp

Third-Party Notification: CERT

The CERT Coordination Center (CERT/CC) is a federally funded organization that provides information on how to protect your systems from attacks, how to react to current attacks, and how to prepare for future problems. CERT provides training and publishes security alerts. For more information, see the CERT Web site: http://www.cert.org/.

Security Reporting

Implementing an effective patch management process requires that your environment be consistently scanned for vulnerabilities. Ideally, you should develop and implement mechanisms for automatically collecting and analyzing information generated by vulnerability scans. Doing so will assist you with properly reacting to security alerts, and efficiently maintaining secure operating baselines.

This chapter includes techniques that can be used to help with security reporting in WU, SUS, and SMS environments. See the Techniques section at the end of this chapter for more information.

Vulnerability Scanning

When security bulletins are released, scanning your entire environment to determine which computers are missing an update could prove very time-consuming, especially in large environments.

Often, the threat of a vulnerability is greater to particular servers in your environment, or the vulnerability targets specific services. Using asset categorization information established during baselining to scan only those computers that are running the affected services can help you complete scans much faster, therefore facilitating faster patch deployments. For example, if a vulnerability puts messaging servers at risk, you might want to scan your enterprise messaging servers first to ensure they meet their operating baselines because of their importance to your environment.

Virus and Intrusion Detection

Although the topic is outside the scope of this document, virus and intrusion detection activities are critical to the overall security of your environment.

All computers in your environment should be evaluated to determine how they can best be protected by a third-party virus protection solution. Ideally, any virus protection software running in your environment should be able to update its signature files through some automated process. As a best practice, periodically check the signature files on sample computers to make sure they address the Microsoft virus alerts described in the Subscription section of this chapter.

Most intrusion detection methods and tools work on the assumption that an intruder's activity is noticeably different than the usual behavior of a regular user. These tools generally analyze log files searching for patterns of suspicious activity. Although these tools often alert you of an attack after the attack has taken place, they can help prevent future attacks by identifying vulnerable systems, and by gathering the necessary forensic evidence to determine the source of an attack.

Microsoft Internet Security and Acceleration (ISA) Server contains features that can assist you with intrusion detection. Information on configuring ISA server for intrusion detection is available at:
http://www.microsoft.com/technet/prodtechnol/isa/proddocs/isadocs/CMT_IntrusionIntro.asp

---

Note: For more information about other virus and intrusion detection tools, see Part I, Appendix A, "Third-Party Tools and Resources."

Reporting a Vulnerability

Microsoft often receives information regarding security vulnerabilities from IT professionals all over the world. Microsoft acknowledges these contributions in the security bulletins it releases. You can review the acknowledgements policy at: https://www.microsoft.com/technet/security/bulletin/policy.asp

In the event that you discover what appears to be a security vulnerability, Microsoft provides a Web form where you can input the required information to report the apparent vulnerability. You will be contacted by a Microsoft support individual to help investigate the potential vulnerability. You can access the Web form at: https://www.microsoft.com/technet/security/bulletin/alertus.asp

Techniques for Setup

Summary

List of Setup Techniques	WU	SUS	SMS
Configuration Techniques			
Remotely Configuring the AU Client Using REG.exe	Yes	Yes	
Baselining Techniques			
Using MBSA to Scan for Computers and Vulnerabilities	Yes	Yes	
SMS Hardware and Software Inventory			Yes
Finding Applications by Using SMS Queries			Yes
Security Reporting Techniques			
Creating a Vulnerability Scanning Report Using MBSA	Yes	Yes	
Creating a Vulnerability Scanning Report Using the SMS Web Reporting Tool			Yes
Using the SMS Distribute Software Updates Wizard for Vulnerability Scanning			Yes

Configuration Techniques

Remotely Configuring the AU Client using REG.exe

Environment:	WU ☑	SUS ☑	SMS ☐
Purpose:	To remotely configure AU clients.		
Prerequisites:	Administrative access to client computers; Resource Kit tools.		
Scale:	The scale of this technique is limited to small and medium-sized environments.		

You can use the REG.exe tool to remotely modify a computer registry. The AURegConfig.cmd.txt file that accompanies this guide contains sample commands for making the necessary configuration changes for the AU client on a remote computer.

Before using this file, you will need to rename it to AURegConfig.cmd, and use a text editor such as Notepad to edit the commands with the configuration parameters that you want. Review the brief instructions included in the file before editing the configuration parameters.

To execute the file, open a command prompt, browse to the folder that contains the AURegConfig.cmd file, and run the following command:

```
AURegConfig.cmd computer name
```

The commands to modify the registry settings on the specified computer will execute and configure the AU client accordingly.

Note: For a detailed description of all the AU client registry settings, see the "Configuring Automatic Updates Client Software" section of the Software Update Services Deployment white paper at: http://www.microsoft.com/windows2000/windowsupdate/sus/susdeployment.asp

Baselining Techniques

Using MBSA to Scan for Computers and Vulnerabilities

Environment:	WU ☑	SUS ☑	SMS ☐
Purpose:	To show how MBSA can help capture computer inventory information.		
Prerequisites:	A computer with MBSA installed in its default location, and administrative access to the computers being scanned.		
Scale:	The scale of this technique is limited to small and medium-sized environments if used as-is.		

MBSA includes both a graphical and command-line interface that can perform local or remote scans of Windows systems.

MBSA runs on Windows 2000 and Windows XP systems, and can be used to scan your environment and collect information from computers that can help you identify what computers are on the network, the products that are installed on each computer, and any applicable security patches or service packs for each identified product.

The MBSAScan.wsf.txt file that accompanies this guide uses MBSA to scan your environment and generate a tab-delimited file that contains detailed information on the computers it can find. The output text file can be examined to determine what specific products are installed on each computer scanned, and provide details related to missing security patches and service packs for each product found.

To use the file, copy it to your computer, and rename it MBSAScan.wsf. To execute the file, open a command prompt, and browse to the folder that contains the MBSAScan.wsf file. Type one of the following commands, depending on the type of scan you want to perform:

To perform a scan on the local computer:

```
Cscript.exe MBSAScan.wsf
```

To scan multiple computers:

```
CScript.exe MBSAScan.wsf -hf servers.txt
```

Note: To scan multiple computers, list the names of all the computers that you want to scan in the servers.txt file. Then, place the servers.txt file in the same directory as the MBSAScan.wsf file.

To scan all computers in a Windows domain called CORPDOMAIN:

```
CScript.exe MBSAScan.wsf -d corpdomain
```

For more information on using the MBSAScan.wsf script:

```
CScript.exe MBSAScan.wsf -?
```

MBSAScan.wsf generates an output file with a file name that represents the time when the scan was performed in the form of YYYYMMDD_HHMMSS. This naming scheme will allow you to run the scan repeatedly without losing data from a previous scan. The output file will be located in the same directory as the script.

Import the output file into the database of your choice for further analysis. You can then query the database for any computer information. You can also import the output file into Microsoft Excel and use the sorting and filtering capabilities of Excel to examine the data.

For the purpose of baselining, you can analyze the data to determine basic inventory information such as what versions of operating systems and Microsoft Internet Explorer are present in your environment, and how many computers are missing the current service pack for each version.

Note: This script assumes that MBSA is installed in its default location (C:\Program Files\Microsoft Baseline Security Analyzer) on the computer running the MBSAScan.wsf file. Scanning your environment may take anywhere from a few minutes to several hours, depending on many factors such as the number of computers being scanned and your available bandwidth.

Furthermore, the inventory information that this script can collect is limited to the products that are supported by MBSA. For the complete list of products that are supported by MBSA, see Table 4.2: MBSA Scanning Capabilities in Part I, Chapter 4, "Tools and Technologies."

SMS Hardware and Software Inventory

Environment:		WU ☐	SUS ☐	SMS ☑
Purpose:	To configure and maintain the SMS software and hardware inventory components.			
Prerequisites:	A working SMS site that has SMS clients.			
Scale:	Software and hardware inventory scales to all sizes of environments.			

Microsoft Systems Management Server provides a robust client-scanning process for both hardware and software. To inventory patches, you must first enable both hardware and software inventory for the SMS site.

Hardware Inventory

SMS hardware inventory uses the Sms_def.mof file to access and retrieve information from the computer's Windows Management Instrumentation (WMI) Repository and from the computer's registry.

Software Inventory

SMS software inventory utilizes a scanning engine that reads the encrypted headers of files to retrieve information about the application. The information that SMS software inventory returns is the same information that you would get if you right-clicked a file name, clicked **Properties**, and then reviewed the information on the **Summary** tab.

SMS Software and Hardware Inventory Scan

SMS performs a software and hardware inventory scan immediately after the SMS client is first installed, so it is important to make sure that the SMS client installs successfully and completely. Both the hardware and software scanning processes are configurable so that you can modify the frequency of the scans on the computer. Most companies perform a hardware scan once a month and run a software scan once a week. The hardware in a computer generally doesn't change too often, but software can be updated on a regular basis if you allow users to install their own.

SMS collects software and hardware inventory information and stores it in a centralized SQL Server database. Those users who have access to the database can review the data, perform queries, generate reports, and create collections of computers based on the search criteria. These collections are used for targeting specific computers for distribution of software and updates.

When you install SMS with the SUS Feature Pack, additional content is added to the Sms_def.mof file that allows the SMS hardware inventory process to retrieve registry information from computers. Patch and update information is stored in the registry when they are installed. This is the same information that you can find in the **Add/Remove Programs** item in Control Panel. The data that is inventoried from **Add/Remove Programs** is stored in the SQL Server database so that you can determine which computers have an update and which do not.

Note: When deploying patches and updates, you should consider changing the schedule for your hardware and software inventory processes before the rollout. The distribution will be more successful if you know that the inventory data for the computers is up to date. Before a large deployment of a critical update, change the inventory schedule so that all computers report their updated inventory prior to the scheduled rollout. The Distribute Software Updates Tool will check computers to ensure that no patches are reapplied after computers have been restarted.

To perform inventory on an as-needed basis, the SMS administrator should create a mandatory advertisement that forces hardware and software inventory to run on a client. The advertisement might also need to run the Security Update Inventory Tool or the Office Update Inventory Tool, which are supplied in the SMS Software Update Services Feature Pack, to identify computers that have yet to install specific patches.

Clients should start running the advertisement within 60 minutes (by default) of when the advertisement reaches their local client access point. The required inventory information should be reported back to the site server shortly thereafter.

You can also force the clients to initiate a hardware and software inventory by using the following method:

1. Create a package that contains only the Cliutils.exe utility from the resource kit.
2. Create two programs:
   - Run hardware inventory. At the command line, type:

     ```
     <driveletter>\<path>\cliutils.exe /START "hardware inventory agent"
     ```

   - Run software inventory. At the command line, type:

     ```
     <driveletter>\<path>\cliutils.exe /START "software inventory agent"
     ```

3. Create two advertisements, using each program line, and then advertise to all clients as soon as possible, using administrative privileges, regardless of whether a user is logged on.
4. Confirm that the advertisement has run successfully.
5. Check the date and time that the last hardware and software inventory was performed on the test clients.

Note: If your SMS site currently has trouble performing hardware and software inventory due to capacity issues, increasing the frequency of inventory is not recommended.

Basic Asset Tracking

Accurate and current knowledge of what is present in the environment is essential for maintaining a smooth-running patching management process. SMS hardware inventory retrieves specific hardware information by default, such as computer drive information, video card attributes, and RAM amounts, and it can be extended to obtain details of installed software patches and other information that is required to support the end-to-end patch management process.

Hardware inventory should be configured to occur daily for data center servers. A weekly inventory should be sufficient for all other computers, unless you are deploying a critical patch. In this event, you will want to modify the hardware inventory schedule to collect inventory as soon as possible without generating an SMS site backlog.

The Sms_def.mof file, which defines what hardware attributes are collected from client computers during hardware inventory, can be found in the SMS\inboxes\clifiles.src\hinv folder on each site server.

Effective patch management requires accurate and current knowledge of what software has been installed in your environment. SMS software inventory retrieves information about every .exe file installed on client computers. Additional work will be required to analyze this inventory to determine which products have actually been installed. In general, software inventory should be configured to occur weekly for all computers in the production environment. Your requirements may be different.

Finding Applications by Using SMS Queries

Environment:	WU ☐	SUS ☐	SMS ☑
Purpose:	To manipulate the data automatically retrieved by SMS using the SMS WMI Query Language (WQL).		
Prerequisites:	A working SMS site, SMS clients, and existing data in the SMS database for inventory.		
Scale:	SMS scales to any size environment.		

Queries are the key to mining the data that is collected by SMS. By using queries, you can extrapolate useful information from a mountain of data. By utilizing queries, you can determine which systems require specific updates. The queries in this section provide examples of how to utilize the query language to create collections for patch deployment targeting. These queries can be used to directly target patches by using computer collections and advertisements.

There are several SMS query examples included with the Tools and Templates that accompany the guide:

- All Systems Running a Prompted Service SMS Query.txt. This SMS query prompts for a service name running on the inventoried computers and then returns all computers that meet the criteria. When prompted, type the name of the service. For example, for the IIS Web service you would type W3SVC.
- All Systems with IIS Installed SMS Query.txt. This SMS query retrieves those computers that have Internet Information Services (IIS) installed.
- All Windows 2000 Professional Workstations SMS Query.txt. Use this SMS query to retrieve a list of all Windows 2000 Professional workstations in your environment.
- All Windows 2000 Servers SMS Query.txt. This SMS query displays all Windows 2000 Servers that have been inventoried by SMS.
- Last Inventory Scan SMS Query.txt. This SMS query prompts for a date and then returns the last time that inventory was run. This information is valuable in determining which SMS clients to force an inventory on before deploying a patch. The process not only requires an accurate inventory before hand, but also requires that inventory be updated post patch installation
- Show All Known Software Products by Company SMS Query.txt. Use this query to list all installed software products, sorted by company name.

Getting Query Results Outside of SMS

There is no command-line function or utility to view the data from SMS queries outside of the SMS Admin console. But, if you need to get the information quickly and don't have immediate access to the SMS Admin console, you can use Microsoft Visual Basic®, Scripting Edition (VBScript) to query the SMS database and display the information.

SMS WMI.vbs (remove **.txt** from the end of the filename), which is provided in the SMS directory within Tools and Templates that accompany this guide, queries the SMS database and then displays the information inside a command prompt. You will need only to supply your SMS site code, and insert the SMS query in place of the query example already in the code.

SMS Excel.vbs (remove **.txt** from the end of the filename), also included in the SMS directory, inserts the retrieved data into an Excel spreadsheet in which the data can be viewed, filtered, and parsed. You only need to insert your own SMS query, replace *SMSSERVER* with the NetBIOS name of your SMS server, and input your SMS site code.

Identifying Computer Types

Identifying computer types is an important part of the asset-tracking process. Knowing whether a computer is a laptop or a desktop can help you pinpoint specific types of computers to which to deploy patches, as well as dictate which post-patching procedures to follow.

Identifying Computer Types MOF.txt, also provided in the SMS Tools and Templates directory, can identify the different types of computers that are deployed in an organization. Add the text from this file to the Sms_def.mof file on *every* site server on which hardware inventory has been enabled. To modify the Sms_def.mof file, browse to \SMS\Inboxes\Clifiles.src\Hinv on the server and edit the file in any text editor. After you have tested the file, you can replace the default Sms_def.mof by replacing the file in the \SMS\Inboxes\Clifiles.src\Hinv directory on the site server. From there, the file is propagated to each client in your site.

Note: You should never edit the site server Sms_def.mof file directly. Instead, work with an offline copy and always make a backup of the file.

After SMS clients have successfully processed the file, the Win32_SystemEnclosure WMI class will be reported to the SMS site server as part of hardware inventory. This information will eventually appear in SMS Resource Explorer and SMS administrators will be able make use of it to build queries and reports.

This code retrieves the manufacturer of the computer, the serial number, and the Chassis Type. The Chassis Type contains the computer type information. The following information is available within the Chassis Type. The values are:

1 = Other
2 = Unknown
3 = Desktop
4 = Low Profile Desktop
5 = Pizza Box
6 = Mini Tower
7 = Tower
8 = Portable
9 = Laptop
10 = Notebook
11 = Hand Held
12 = Docking Station
13 = All in One
14 = Sub Notebook
15 = Space-Saving
16 = Lunch Box
17 = Main System Chassis
18 = Expansion Chassis
19 = SubChassis
20 = Bus Expansion Chassis
21 = Peripheral Chassis
22 = Storage Chassis
23 = Rack Mount Chassis
24 = Sealed-Case PC

Inventorying Internet Explorer

Internet Explorer has been involved in many security-related vulnerabilities since 2000. Special attention should be directed to ensuring that Internet Explorer is properly patched. Internet Explorer inventory is necessary because there are so many different versions in use, each with its own vulnerabilities. Inventorying Internet Explorer is a little more involved than just reviewing the information that is collected by SMS for the Iexplore.exe file through the standard inventory process. The only accurate way to retrieve Internet Explorer version information is to cause SMS to inventory the registry key on the computer that holds the Internet Explorer data. The registry keys contain information that SMS can use to inventory the version, service pack level, and hotfixes applied. To allow SMS to inventory the registry key, you must modify your Sms_def.mof file to include a Registry Provider (a bit of code that gives SMS access to the registry) and then new information to tell SMS which registry information to retrieve. This registry provider code is included in Registry Provider MOF.txt, included in the SMS file in Tools and Templates.

For more information about implementing the WMI Registry Providers, see the Extend Sms_def.mof by Using WMI Registry Providers white paper at: http://www.microsoft.com/smserver/techinfo/administration/20/using/extenddefmof.asp.

Internet Explorer Code

Internet Explorer Inventory MOF.txt, also provided in the SMS file in Tools and Templates, needs to be copied into your Sms_def.mof file. Make sure it follows the Registry Provider code. This bit of code inventories the actual Internet Explorer information.

Security Reporting Techniques

Creating a Vulnerability Scanning Report with MBSA

Environment:	WU ☑ SUS ☑ SMS ☐
Purpose:	To determine how many computers are vulnerable to a specific attack.
Prerequisites:	A computer with MBSA installed in its default location, and administrative access to the computers being scanned.
Scale:	The scale of this technique is limited to small and medium-sized environments.

MBSA includes both a graphical and command-line interface that can perform local or remote scans of Windows systems.

MBSA runs on Windows 2000 and Windows XP systems, and can be used to scan your environment and collect information from computers that can help you identify what computers are on the network, the products that are installed on each computer, and any applicable security patches or service packs for each identified product.

The MBSAScan.wsf.txt file that accompanies this guide uses MBSA to scan your environment and generate a tab-delimited file that contains detailed information on the computers it can find. The output text file can be examined to determine what specific products are installed on each computer scanned, and provide details related to missing security patches and service packs for each product found.

To use the file, copy it to your computer, and rename it MBSAScan.wsf. To execute the file, open a command prompt and change to the folder that contains the MBSAScan.wsf file. Type one of the following commands, depending on the type of scan you want to perform:

To perform a scan on the local computer:

```
Cscript.exe MBSAScan.wsf
```

To scan multiple computers:

```
CScript.exe MBSAScan.wsf –hf servers.txt
```

Note: To scan multiple computers, list the names of all computers that you want to scan in the servers.txt file. Then, place the servers.txt file in the same directory as the MBSAScan.wsf file.

To scan all computers in a Windows domain called CORPDOMAIN:

```
CScript.exe MBSAScan.wsf –d corpdomain
```

For more information on using the MBSAScan.wsf script:

```
CScript.exe MBSAScan.wsf -?
```

MBSAScan.wsf generates an output file with a file name that represents the time when the scan was performed in the form of YYYYMMDD_HHMMSS. This naming scheme will allow you to run the scan repeatedly without losing data from a previous scan. The output file will be located in the same directory as the script.

Import the output file into the database of your choice for further analysis. You can then query the database for any computer information. You can also import the output file into Excel and use the sorting and filtering capabilities of Excel to examine the data.

For the purposes of creating a vulnerability report, you can sort or filter the output file to include only those computers that are missing a specific security patch.

Note: This script assumes that MBSA is installed in its default location (C:\Program Files\Microsoft Baseline Security Analyzer) on the computer running the MBSAScan.wsf file. Scanning your environment may take anywhere from a few minutes to several hours, depending on many factors such as the number of computers being scanned and your available bandwidth.

Furthermore, the vulnerability information that this script can collect is limited to the products that are supported by MBSA. For the complete list of products that are supported by MBSA, see Table 4.2: MBSA Scanning Capabilities in Part I, Chapter 4, "Tools and Technologies."

Creating a Vulnerability Scanning Report Using the SMS Web Reporting Tool

Environment:	WU ☐	SUS ☐	SMS ☑
Purpose:	To determine how many computers are vulnerable to a specific attack.		
Prerequisites:	A working and healthy SMS site, inventory processes, and status system; the Web Reporting tool installed on an IIS server.		
Scale:	Scales to all sizes of environments.		

After you install the SUS Feature Pack along with the Web Reporting tool, SMS will automatically report inventory for installed patches. The Web Reporting tool provides the stock reports shown in the following table to help you determine the scope of a vulnerability.

Table 1.2: Patch Management Reports for the Web Reporting Tool

Report	Description
Installed software updates for a Specific Machine	Use this report to obtain a list of installed software updates for a single computer.
Count of installed Software Updates by type	Use this report to obtain an enterprise-wide count of installed software updates by type.
Installation rate for a specific software update	Use this report to obtain enterprise-wide information about the rate of installation for a software update.
Installed software updates for a specific product	Use this report to obtain an enterprise-wide list of all installed software updates for a specific product.
Machines with a specific software update installed	Use this report to obtain an enterprise-wide list of computers with a specific software update installed.
Machines with any software update installed	Use this report to obtain an enterprise-wide list of computers with any software updates installed.
Applicable software updates for a specific machine	Use this report to obtain a list of all applicable software updates for a specific computer.
Count of applicable software updates by type	Use this report to obtain an enterprise-wide count of applicable software updates by type.
Detection rate for a specific software update	Use this report to obtain an enterprise-wide list of all applicable software updates for a specific product.
Applicable software updates for a specific product	Use this report to obtain an enterprise-wide list of all computers for which a specific software update is applicable.
Count of applicable software updates by type	Use this report to obtain an enterprise-wide list of all computers for which any software update is applicable.
Detection rate for a specific software update	Use this report to view enterprise-wide status counts for all software updates, both authorized and unauthorized.
Applicable software updates for a specific product	Use this report to obtain an enterprise-wide status counts for all authorized software updates.
Machines where a specific software update is applicable	Use this report to obtain an enterprise-wide list of all authorized software updates, including site, computer, and installation status for each.
Machines where any software update is applicable	Use this report to view computers in your enterprise that have a number of applicable software updates greater than or equal to a value that you select.

Using the SMS Distribute Software Updates Wizard for Vulnerability Scanning

Environment:	WU ☐	SUS ☐	SMS ☑
Purpose:	To use the Distribute Software Updates Wizard to provide information for determining the vulnerability state of the environment.		
Prerequisites:	A working SMS site with the SUS Feature Pack installed; healthy software and hardware inventorying processes.		
Scale:	SMS scales to any size environment.		

When administrators use the Distribute Software Updates Wizard, they can specify a collection that automatically creates a repeating advertisement running the patch installation agent. The repeat interval can be altered from the default (seven days), as appropriate for the collection. For example, the collection that includes the servers may run the agent once per day or even more often.

If different schedules are needed for different types of computer, multiple advertisements can be created for multiple collections, using the same package and program.

If the repeat interval for the running of the patch installation agent is set to daily and it needs to be run sooner for the rollout of a critical patch, a new, one-time, mandatory assignment should be made for the advertisement to run as soon as possible.

As soon as the advertisement changes have propagated to the client access points, the next advertised programs check on the client will run the patch installation agent and install the new patch.

If the Distribute Software Updates Wizard is not being used, but the patches are being distributed through a custom package and collection, a single advertisement should be created to run the patch installation, using the collections created in the Select Deployment Group section.

If you are staggering the rollout and using a single query/collection, when rollout of the first phase is complete, the query should be modified to include clients in the next phase, for example the next site(s) to be deployed. The collection is either updated manually or on a defined schedule, after which the patch installation program is automatically advertised to the new set of clients.

For hotfixes, the SMS queries used for the target collections are based on inventory information indicating which computers do not have the patch that is being installed. This means that:

- As computers install the patch and are subsequently re-inventoried, these computers will drop out of the collection when the collection updates because they will no longer match the query definition.
- When a new computer is added to the network and inventoried by SMS and when the collection updates, if the patch is not on that new computer, the patch installation program will automatically be advertised to it.

The repeat interval of such an advertisement needs to be chosen carefully because:

- Clients that have installed the release successfully will remain in the target collection until a new inventory has been collected (which may be triggered by the installation program itself) or until the inventory has been updated in the SMS database and the collection is reevaluated. Therefore, the program could be rerun on a client that has already successfully run it once. For this reason, it is essential that the program to be run checks first to see if the patch or patches are installed and exits without delay if they are.
- Repeated running of a program that always fails because of some other error will become annoying to users.
- Repeated running of an installation program creates additional client load and network load.
- The interval must, however, not be too long, particularly when the need to apply a patch is urgent.

There are situations in which a patch is critical and should be deployed in a very short timeframe. Because this will usually come under the priority of an urgent change, the test phase will probably have been very short. Because of the increased risk involved, the target query and collection should be kept to the minimum number of clients possible, bearing in mind the risk of not deploying the patch.

The SMS architecture, in which all servers in the data centers are in a single SMS site, allows the critical patch to be deployed in the shortest possible time. After the package is created at that site, the target collection established and the program advertised, there should be no delays in this information reaching the server SMS clients due to the SMS architecture chosen. The advertisement will be run as soon as the servers next check for new advertisements, which means that the interval set for this should be short.

3

Change Initiation

Summary

The change initiation process for patch management has three major components:

- **Identification.** Determining whether a patch is required by your environment, and whether its source is valid.
- **Relevance.** Determining whether a patch is meaningful within the context of your organization's IT infrastructure.
- **Quarantine.** Isolating any files related to a patch or patches while they are examined for viruses or other malicious code that might affect your organization's IT infrastructure.

The following flow chart illustrates the change initiation process:

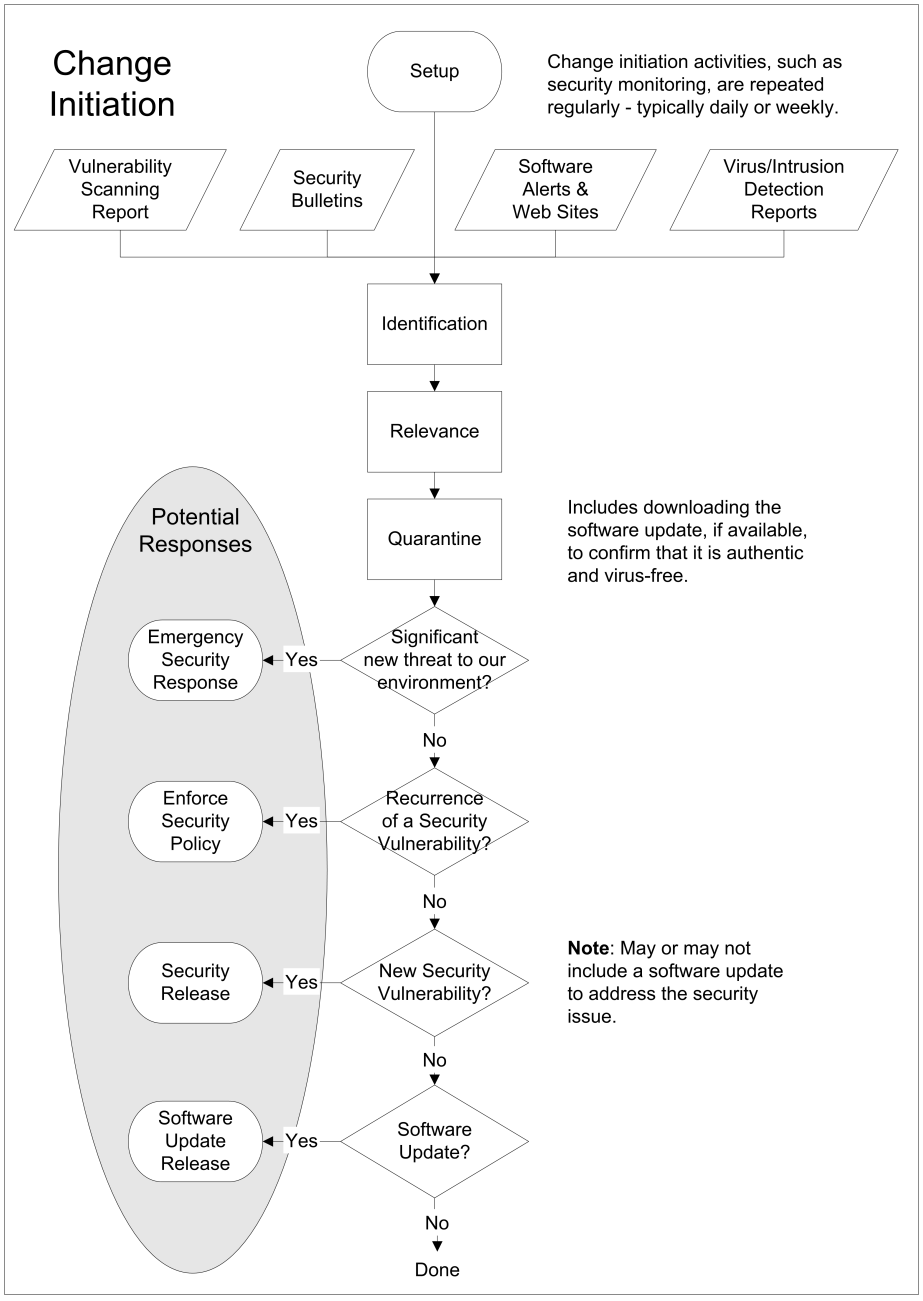


Figure 2.1
Change Initiation for Security Patch Management

Identification

Proper monitoring and analysis of your environment's vulnerability scanning reports, as well as validating the source and content of security bulletins that you receive are essential to initiating your patch management process.

Vulnerability Scanning Reports

Proactively analyze the results of vulnerability scans performed in your environment to identify potential vulnerabilities. Reports based on the different categories of computers or specific technologies running in your environment can help you react quickly to vulnerabilities.

Microsoft® Baseline Security Analyzer (MBSA) is a useful tool for vulnerability scanning in the Microsoft Windows® Update (WU) and Microsoft Software Update Services (SUS) environments. Microsoft Systems Management Server (SMS) with the SUS Feature Pack includes MBSA

Microsoft Security Bulletins

E-mail notifications include details on the vulnerabilities they report. Closely review all information included in the e-mail notifications. However, you should always refer to the Microsoft Security Web site as the most comprehensive and up-to-date source of information on security bulletins at http://www.microsoft.com/technet/security/default.asp.

Follow these guidelines when reviewing the contents of all e-mail notifications to validate the notification, and to ensure you obtain the latest security bulletin information available:

- Immediately delete any e-mail notifications claiming to be from Microsoft that contain any attached software files. Never run or install any executable attached to an e-mail notification.
- Do not click any links directly from inside an e-mail notification. Instead, you should paste any URLs into a browser window to confirm they direct you to a Microsoft Web site.
- Always visit the Microsoft Security Web site to read the authoritative details of a security bulletin. Alternatively, if you don't expect to have Internet access when receiving bulletins, familiarize yourself with PGP tools and select one that you can use to verify the PGP signature included on each security bulletin to confirm its authenticity.

  You can download the Microsoft Security Response Center (MSRC) Security Bulletin key from: http://www.microsoft.com/technet/security/MSRC.asc.

  For more information about how to verifying the digital signature, see: http://www.microsoft.com/technet/security/bulletin/notify.asp.

There are several e-mail hoaxes that claim to be from Microsoft. When you receive a Microsoft Security bulletin, confirm it and *all hyperlinks to software updates* by visiting the Security Bulletin Search Tool Web site at: http://www.microsoft.com/technet/security/current.asp.

For more information about these types of hoaxes, see Information on Bogus Microsoft Security Bulletin E-mails: http://www.microsoft.com/technet/security/news/patch_hoax.asp.

Note: Microsoft has a policy of never distributing software through e-mail attachments. Please review the Microsoft Policies on Software Distribution at: http://www.microsoft.com/technet/security/policy/swdist.asp

Reading a Microsoft Security Bulletin

Information in security bulletins on the Microsoft Web site is arranged into sections that help you determine how critical the described vulnerabilities are to your environment. Although you should review all information in a security bulletin, pay close attention to the following items when you first examine a security bulletin:

- **Summary.** Immediately review the Summary section of a security bulletin. The Maximum Severity Rating, Impact of vulnerability, Affected Software, and Recommendation items contain information that will assist you with determining how susceptible your environment is to a vulnerability.
- **Technical Details**. The Technical Details section provides in depth technical description of the vulnerabilities in a security bulletin. It also outlines the mitigating factors and the severity of the vulnerabilities for all affected products.
- **Knowledge Base article.** Reference the Knowledge Base article identified in the title of a security bulletin. Additional information on the vulnerabilities and any updates prescribed by the security bulletin is provided in the Knowledge Base. The number in parentheses to the right of a security bulletin's title indicates the security bulletin's corresponding Knowledge Base article. Use this number to search for the article on the Microsoft support Web site at http://www.support.microsoft.com

For example, a security bulletin could include the following information in its Summary section:

- **Impact of Vulnerability:** Allow an attacker to execute code on a user's system.
- **Maximum Severity Rating:** Critical
- **Recommendation:** Customers should install this patch at the earliest opportunity.
- **Affected Software:**
  - Microsoft Internet Explorer 5.01
  - Microsoft Internet Explorer 5.5
  - Microsoft Internet Explorer 6.0
  - Microsoft Internet Explorer 6.0 for Windows Server 2003.

The brief descriptions in the Summary section allow you to make a quick assessment of the potential impact a vulnerability might pose to your environment without having to closely review the entire contents of a bulletin. The Summary highlights the type of exploit, provides a list of the affected software, and recommends the proper course of action. After reviewing the Summary section, you should be able to determine if you need to immediately perform an in-depth review of the remaining section of the security bulletins.

Maximum Severity Rating System

The Microsoft Security Response Center (MSRC) has implemented the Maximum Severity Rating System to assist you with quickly determining how important an update is to your organization. These ratings are based on the potential impact of a vulnerability, and are intended to inform you of the urgency of any required actions. Updates can be assigned any of the following severity ratings:

- **Critical.** A vulnerability whose exploitation could allow the propagation of an Internet worm without user action.
- **Important.** A vulnerability whose exploitation could result in compromise of the confidentiality, integrity, or availability of users' data, or of the integrity or availability of processing resources.
- **Moderate.** A vulnerability whose exploitation is mitigated to a significant degree by factors such as default configuration, auditing, or difficulty of exploitation.
- **Low.** A vulnerability whose exploitation is extremely difficult, or whose impact is minimal.

Note: For more information on the Maximum Severity Rating System, see: http://www.microsoft.com/technet/security/policy/rating.asp

---

Relevance

A large number of software updates are regularly released into the IT operations community. These software updates can come from many sources and have been created for many reasons, including addressing problems that could lead to potential security breaches. Review patches carefully, but do not immediately assume they must be deployed to your environment. With so many patches available for so many different kinds of software, it is crucial to ensure all patches are evaluated thoroughly for their relevance to your environment to maintain the most efficient patch management process possible.

Not every security patch that Microsoft releases will be relevant to your environment. Although it is important to be aware and have a good understanding of existing security patches, deploying only those patches that are relevant to your environment will minimize the effort that is required to keep your environment up to date and secure.

From the information gathered in the initial notification and the further information discovered in checking the patch for relevance, you should be able to ascertain if the software update is applicable to your organization.

Note: Determining the relevance of technology-specific patches can pose significant challenges to any environment. Technologies, such as the Microsoft Date Engine (MSDE) or Microsoft Internet Information Services (IIS), that can be installed on clients or servers can make it difficult to determine if a software update is relevant to any specific subset of computers in your environment. This emphasizes the importance of maintaining as accurate an inventory of your environment as possible.

Quarantine

After you have determined that a patch is relevant to your environment, the next step is to download any related files. On rare occasions, security patches will only require you to make registry or configuration file changes or application settings, but most security patches will involve downloaded files. Always quarantine the files that you download by isolating them from your production network while you examine them for viruses and confirm their digital authenticity to endure they won't negatively affect your organization's IT infrastructure.

Note: The Microsoft Download Center, Windows Update (and the Windows Update Catalog), SUS, and SMS with the SUS Feature Pack only provide authentic, Microsoft software updates. If you receive a Microsoft software update through any other means, take efforts to confirm its validity and digital signature.

Initial installation and testing of a security patch should also be performed on computers that are isolated from your production network. Ideally, to ensure that strict controls are in place in the quarantined environment, the quarantine process should be carried out by a dedicated group within the organization.

The following sections describe a few ways that you can download software updates for alternate distribution mechanisms and installation purposes outside of using the built-in capabilities of the Automatic Updates client (for WU and SUS) and SMS with the SUS Feature Pack.

Downloading Updates from the Microsoft Download Center

To download a software update posted by Microsoft on the Download Center:

1. Visit the Microsoft Download Center: http://www.microsoft.com/downloads.
2. In the **Keywords** box, type in the Knowledge Base (KB) article of the software update. You may have to put a KB or Q in front of the number and attempt the search a couple of times. The Microsoft security bulletin number is not used as a software update keyword. Alternatively, select a specific product from the **Product/Technology** drop-down list and then click **Go**.
3. Click the software update that is appropriate for your operating system or application.
4. Review the information provided on the download page for that software update to be sure you are downloading the correct one.
5. If necessary, select a language from the drop-down list on the right-hand side of the page and click **Go**.
6. Click on the Download link or follow the instructions on the download page.

Note: When downloading a software update from www.microsoft.com, it is always best to type in the Download Center Web site URL directly and then search for the download manually. This prevents you from following the URL of a spoofed message that might take you to a site that is not authentic.

Downloading Updates from the Windows Update Catalog

You can access the Windows Update Web site in the following ways:

- Using Internet Explorer, on the toolbar, click **Tools** and then click **Windows Update**.
- In Windows, click the **Start** button, and then click the **Windows Update** icon.
- Visit the Windows Update Web site: http://windowsupdate.microsoft.com.

Note: For Windows Update to work properly, you must change the security settings for the Internet zone to **Medium** or lower. To do so, go to Internet Explorer, click the **Tools** menu, and then click **Internet Options**. Click the **Security** tab, and then click the **Internet** icon. To adjust your security level, click **Default Level** or **Custom Level**.

To view the Windows Update Catalog, you have to add the Windows Update Catalog to your list of choices on the default Windows Update page. To access this catalog, perform the following steps:

1. Under **Other Options**, click **Personalize Windows Update**.
2. Select the check box for the **Display the link to the Windows Update Catalog under See Also** option.
3. Click **Save Settings**.
4. Under **See Also**, select the **Windows Update Catalog** link.
5. Perform a search by operating system and language, optionally choosing one or more update types from the **Advanced search** options.

Note: The Windows Update Catalog currently categorizes security patches in the Critical Updates and Service Packs category.

Downloading Updates from the Office Download Center

To download a software update by finding it through the Office Download Center:

1. Visit the Microsoft Office Download Center: http://office.microsoft.com/Downloads.
2. Use the Product and Version drop-down boxes to choose a product and then select only the **Updates** check box.
3. Click **Update List**.
4. Click the **Download Now** link for the software update you are looking for.

4

Security Release

Summary

The security release process for patch management has three major components:

- **Change management.** Outlines a set of procedures and disciplines that are designed to reduce the number of unnecessary changes to your environment, and to ensure that no service issues or outages are caused by any changes.
- **Release management.** Provides the steps for planning, testing, and delivering changes to your production environment while maintaining the integrity and availability of any existing services.
- **Change review.** Consists of the steps that are required to determine the success of a patch deployment, and provides considerations for stopping and rolling back the deployment in the event of an unsuccessful deployment.

The following flow chart illustrates the complete security release process:

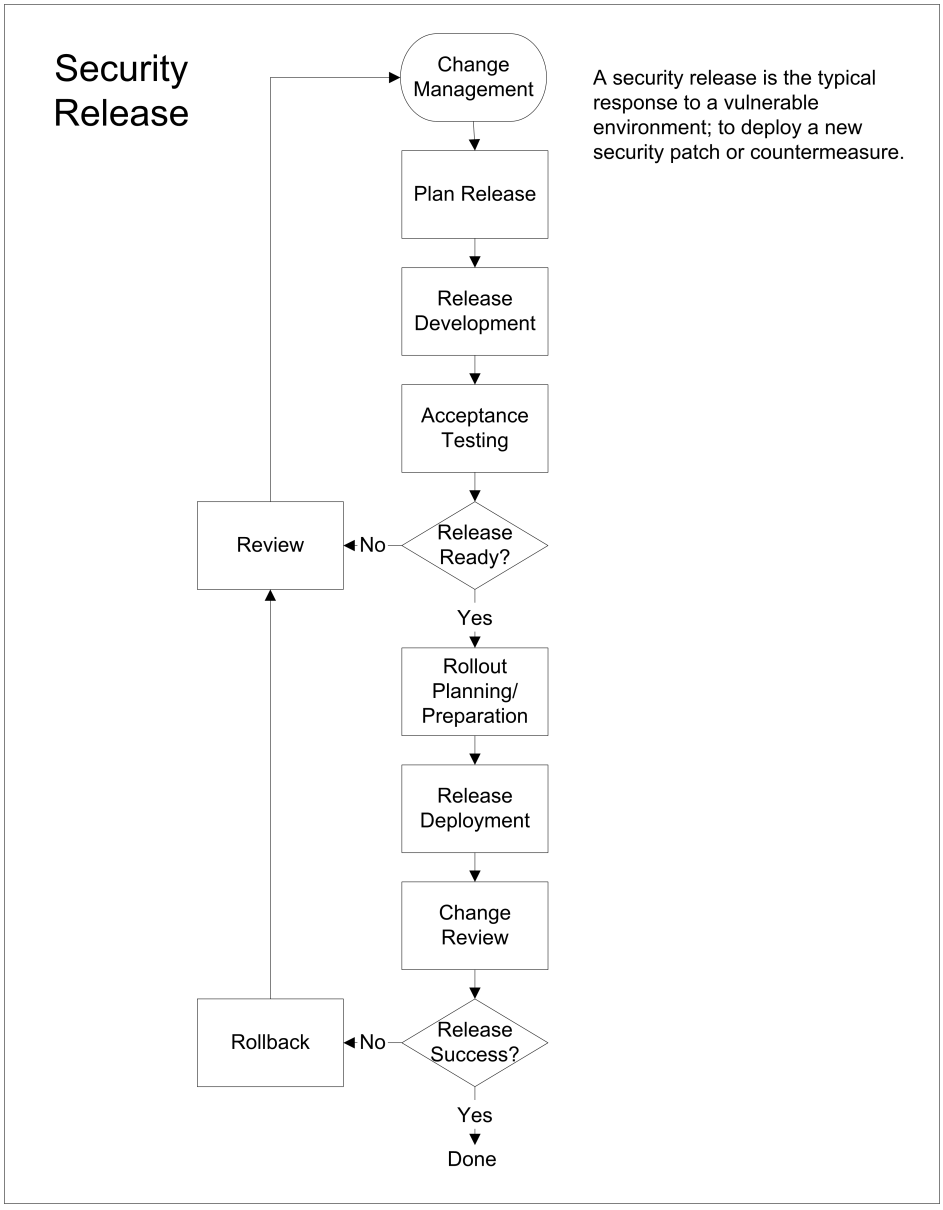


Figure 4.1
Security Release for Security Patch Management

Change Management

Change, in the context of patch management, is defined as any additions to software, such as software updates, that are deliberately introduced into an IT environment and that might impact the proper functioning of the environment or one of its components.

Changes can be either temporary or permanent. Temporary changes, such as countermeasures, may be put in place until a permanent change, most likely a software update, can be implemented. Whether temporary or permanent, all changes should be managed by the change management process.

As security bulletins are released, and relevant patches are validated for deployment, it is important to identify the resources in your environment that are most affected by the vulnerability, as well as to identify which of these resources are most critical to the proper operation of your environment. The asset categorization information established earlier will assist you with finding the resources that are most critical to your environment so you can deploy software updates to those computers as quickly as possible. A well-defined change management process will allow you to expedite the approval of any required security patches for these computers.

The Tools and Templates that accompany this guide includes Request for Change Form.doc, a sample form that can assist you with the change management process. The form helps you identify and organize the most important information regarding a security patch.

Note: Patch change management is only part of a complete change management process. The introduction of changes to any area of your IT environment should adhere to the same change management process. For best practice information regarding change management, see:
http://www.microsoft.com/technet/ittasks/maintain/pracserv.asp

---

Risk Assessment

Before deploying a security patch, you must balance the risk of the vulnerability with the risks involved in addressing it. For example, if the security patch requires your computers to restart, you might need to schedule the deployment during non-business hours, or, at a minimum, you might need to stagger the deployment to reduce the impact of the computer restarts.

The following is a list of a few considerations that will assist you with properly assessing the risks posed by the deployment of a security patch.

- **Determine the exposure a particular vulnerability poses to your environment.** Many security bulletins and related patches may apply only to a few computers in your environment. If the threat of the vulnerability is low, the deployment of any related patches can be scheduled to occur during the next maintenance window.
- **Prioritize releases based on asset valuation.** Give special consideration to computers running services that are crucial to the proper operation of your environment. You should be aware of how any downtime resulting from the release will impact your environment.
- **Network infrastructure impact.** Deploying a large patch to many computers simultaneously could degrade network performance and adversely affect the proper operation of your entire environment. Closely review all software update documentation, and always be aware of patch size and the number of computers that will receive the patch. This information can assist with properly scheduling the release.
- **Determine the impact of computer downtime.** Ensure you are aware of the repercussions of downtime in the rare event that a patch causes a computer to malfunction. You will need to compare and consider the risks of postponing a patch deployment versus the risks incurred by causing computer downtime when deploying a patch to your environment.

Note: Even the smallest of IT failures has the capacity to severely cripple your environment. Understanding the risks posed by changes you plan to make to your environment is an essential part of an effective change management process. For a detailed overview of risk assessment best practices, see: http://www.microsoft.com/technet/itsolutions/tandp/opex/mofrl/mofrisk.asp.

Classification

Classification is the change management step in which you determine the priority of a release and its resulting schedule.

The following table provides some guidelines for various security release priorities.

Table 4.1: Security Release Deployment Timeframe Guidelines

Priority	Recommended Timeframe	Minimum Recommended Timeframe
1	Within 24 hours	Within 2 weeks
2	Within 1 month	Within 2 months
3	Depending on availability, deploy a new service pack or update rollup that includes a fix for this vulnerability within 4 months.	Deploy the software update within 6 months
4	Depending on availability, deploy a new service pack or update rollup that includes a fix for this vulnerability within 1 year.	Deploy the software update within 1 year, or may choose not to deploy at all.

The Microsoft® Security Response Center (MSRC) rates the severity of the vulnerabilities described in the security bulletins it releases in an effort to assist customers with deciding which patches to apply to avoid impact under their particular circumstances, and how rapidly they need to take action.

The priority and resulting schedule for a security release should be determined by taking into consideration the defined MSRC severity level of the vulnerability along with aspects that are unique to your environment.

A simple mechanism for determining priority involves mapping the MSRC-defined severity level of the vulnerability to an initial priority level for the release (Table 4.2), then raising or lowering the priority level depending on the needs of your organization and unique aspects of your environment (Table 4.3).

Table 4.2: Determining Projected Priority

MSRC Severity Level of Vulnerability	Initial Priority
Critical	1
Important	2
Moderate	3
Low	4

Table 4.3: Factors That May Influence Release Priority

Environmental/Organizational Factor	Possible Adjustment of Priority
High value or high exposure assets impacted	Raise
Assets historically targeted by attackers	Raise
Mitigating factors in place, such as countermeasures that minimize the threat	Lower
Low value or low exposure assets impacted	Lower

To ease the impact on resources performing release management, multiple security changes for the same category of asset can be combined into a single release. This is most appropriate for priorities 2 through 4, as listed in Table 4.1.

Typical Countermeasures

The majority of patches require that target computers be restarted before the installation is complete and the computer is safeguarded. If you are prevented from immediately deploying a patch to your environment because computer restarts are limited to specific maintenance windows, implementing any recommended countermeasures can effectively safeguard your computers until the patch can be deployed.

As software updates are identified and applied to the organization, there are several things to keep in mind in relation to dealing with countering vulnerabilities as the release is distributed.

The security bulletin should provide additional countermeasures to take to deter further attack before the software update is applied. The security bulletin on the Microsoft Web site provides further information about the vulnerability in the Technical Details and Frequently Asked Questions sections. You can use this information to develop a countermeasure plan until the software update is deployed completely.

Note: For an example, review Microsoft Security Bulletin MS03-010: http://www.microsoft.com/technet/security/bulletin/MS03-010.asp.

In the Technical Details section of MS03-010, review the additional information about how the vulnerability can be exploited over the Remote Procedure Call (RPC) port 135. When you expand the Frequently Asked Questions section, information on blocking port 135 as a recommended countermeasure is provided in the Workarounds section.

Furthermore, implementing computer hardening countermeasures can often protect computers from many common vulnerabilities. Blocking certain network ports, and disabling unused services are some of the countermeasures that, when implemented, can effectively safeguard your computers. For more information on computer hardening countermeasures, see: Threats and Countermeasures: Security Settings in Windows Server 2003 and Windows XP at: http://go.microsoft.com/fwlink/?LinkId=15159.

Note: Countermeasures are not replacements for software update deployments. Instead, countermeasures are intended to assist you with securing your environment while a software update is being deployed. Your goal should always be to secure your environment by deploying software updates accordingly.

Release Management

Release management is the process of planning, testing, and executing changes to your environment. Its goal is to ensure that all changes are deployed successfully into production in the least disruptive manner possible. The following sections describe things to consider during the different phases of release management that will assist you with defining the proper process for effective patch deployment.

Plan Release

During the release-planning phase, you need to define deployment details based on how critical the release is to your environment. Effective release planning will require you to make the following determinations:

- **Scope of the release.** Which computers in your environment need the software update? Is your environment's current inventory information up to date? Accurately determining which computers need a specific release contributes to the most efficient deployment possible.
- **Release order for multiple software updates.** What dependencies do any approved updates have? Which require a system restart? Can multiple updates be combined? Answering these questions can help minimize the impact a release has on your environment by ensuring software updates install correctly, and by limiting system downtime caused by unnecessary restarts.
- **Rollback considerations.** Can the release be uninstalled? Are necessary provisions in place in the event a computer stops responding after a patch is deployed? Are the proper data backup and restore procedures taking place? Understanding the requirements for returning computers to their original state in the unlikely event that a deployment adversely affects your environment is an important aspect of release management.
- **Schedule for release deployment.** Does the release require immediate deployment? Can it be phased into production over time? More than likely, not all computers in your environment will require an update immediately. Once again, answers to these questions will help you minimize unnecessary downtime that could adversely affect your environment.
- **Define end-user training and communication requirements.** What should be communicated to end-users and when? Communicating the severity, urgency, and potential adverse effects of a deployment to end-users will assist you with limiting the negative effects of the deployment to your environment. To make sure that end-users understand the importance of a release, you should use multiple communication mechanisms to inform them of software updates that require immediate deployment. Suggested communication mechanisms include broadcast voice mail or high priority e-mail sent by a trusted internal source.

Release Development

With a release plan in place, direct your efforts to identifying and developing the release mechanism used to deploy patches into production. A release mechanism consists of the processes, tools, and technologies that are required for completing a deployment. During this process, you select the release mechanism and design, develop, and test the release mechanism, the release installation, and the results of the release deployment.

The release mechanism that you choose can be greatly influenced by the priority assigned to a patch during the classification process. Releases that need to be deployed quickly might require multiple release mechanisms to ensure deployment.

The following release considerations will assist you with selecting the proper release mechanism. The mechanism that you choose should ensure that the release can be:

- Deployed quickly over the slowest links in your environment.
- Installed remotely without requiring user intervention.
- Configured to allow users to save their work before the system restarts.
- Installed on only those computers that are required to apply the patch.
- Logged with sufficient detail so that post-deployment audits can determine success or failure.
- Integrated into the standard build process.

Note: Read the Knowledge Base (KB) article of a software update carefully. If the software update cannot be uninstalled, ensure the proper provisions are developed for performing a rollback in the event the release interferes with the proper operation of any computers. This may include rebuilding the computer or reinstalling applications.

Understanding Software Update Packages

Microsoft software updates from different product groups may be packaged with different installers to meet the unique needs of each product. Different installers may have different command-line parameters and different capabilities (such as uninstall).

The following resources describe how various software updates are named, which installers the software update packages use, the command-line switches available with each installer, and other related resources.

Note: Microsoft is working to standardize the command-line parameters for all software update packages. Be sure to check the command-line switches that accompany each software update KB article to learn of improvements as they occur.

Many of the following references use the term *hotfix* to refer to any type of software update that is not a service pack; in the new software update taxonomy, a hotfix is a non-public fix—the following resources may not reflect this new taxonomy.

Windows NT 4.0

Software updates for Microsoft Windows NT® 4.0 and Microsoft Windows® 2000 SP3 and earlier are packaged with an installer called Hotfix.exe.

How to Install and Remove Hotfixes with HOTFIX.EXE:
http://support.microsoft.com/?kbid=184305.

Hotfix Packages Do Not Include Debug Symbol Files:
http://support.microsoft.com/?kbid=814411.

Windows 2000, Windows XP, and Windows Server 2003

Software updates for Windows 2000 SP4 and later, Windows XP, and Windows Server 2003 are packaged with an installer called Update.exe. Update.exe is a new version of the Hotfix.exe installer.

New Naming Schema for Microsoft Windows Hotfix Packages:
http://support.microsoft.com/?kbid=816915.

Hotfix.exe Program Description and Command-Line Switches:
http://support.microsoft.com/?kbid=262841.

How to Install Multiple Windows Updates or Hotfixes with Only One Reboot:
http://support.microsoft.com/?kbid=296861.

How to Install Microsoft Virtual Machine Updates Silently Without Restarting Your Computer:
http://support.microsoft.com/?kbid=304930.

Internet Explorer

Software updates for Microsoft Internet Explorer are packaged with an installer call IExpress.exe.

Common Command-Line Switches for Self-Installing Update Files:
http://support.microsoft.com/?kbid=197147.

Office 2000 and Office XP

Software updates for Microsoft Office 2000 and XP are packaged with an installer called Ohotfix.exe. Ohotfix.exe is a launcher for MSP (Windows Installer patch) files, using the Windows Installer to install software updates.

New Naming Schema for Microsoft Office Hotfix Packages:
http://support.microsoft.com/?kbid=816916.

Microsoft Office XP Patch Deployment White Paper:
http://support.microsoft.com/?kbid=330043.

Installing Client Update Files with OHotFix:
http://www.microsoft.com/office/ork/xp/journ/ohotfix.htm.

List of OHotFix Installation Errors:
http://support.microsoft.com/?kbid=324246.

Exchange Server

Software updates for Microsoft Exchange 5.0 and Exchange 5.5 (on Windows NT 4.0) are packaged with the SMS Installer. Software updates for Exchange 2000 use Update.exe except for service packs, which use a custom installer.

New Naming Schema for Exchange Server Hotfix Packages:
http://support.microsoft.com/?kbid=817903.

Exchange 2000 Server Post-Service Pack 3 Hotfix Command-Line Switches:
http://support.microsoft.com/?kbid=331646.

SQL Server 7.0 and SQL Server 2000

Software updates for Microsoft SQL Server™ use a variety of installers depending on the type of software update, the platform, and the version of SQL Server.

SQL Server Hotfix Installer:
http://support.microsoft.com/?kbid=330391.

Visual Studio

Microsoft Visual Studio® 6.0 uses IExpress for software update installers. Visual Studio .NET uses Windows Installer 2.0 with a wrapper for software update packages.

Naming Schema for Microsoft Visual Studio .NET, the .NET Framework, and Visual J# Redistributable Hotfix Packages:
http://support.microsoft.com/?kbid=822464.

Acceptance Testing

The purpose of the testing that you have done to this point is to confirm that the release works correctly within a development environment. Acceptance testing allows your business managers to see your deployment plan and release mechanisms perform together in an environment that closely mirrors production.

Furthermore, acceptance testing needs to demonstrate that your deployment plan will not adversely affect your production environment. In some cases, a pilot deployment to a small group of production users may be required in addition to acceptance testing to build confidence for proceeding with a corporate-wide rollout.

Note: Comprehensive acceptance testing can be very difficult in environments in which not all computers are installed using a similar build standard. It is often not possible to reproduce all of the different computer configurations that might exist in your environment. If this is the case, focus on ensuring that your data backup processes complete as scheduled, and that your rollback procedures can be executed as designed.

To ensure releases will not adversely impact the production environment, each release should be tested in a facility that effectively models the conditions that exist in production. The test environment should also be structured to handle new computers entering below the operational baseline. For example, if a new computer build consists of Windows XP without any patches, your release mechanism should install all necessary software updates to bring the computer up to the established operational baseline.

Note: To accelerate the acceptance testing process, or when it is not possible to test every type of computer in your environment, you can search security-related newsgroups for information pertaining to specific security bulletins and related software updates. Security-related newsgroups can be accessed by configuring your newsgroup reader (for example, Microsoft Outlook® Express) to point to news://msnews.microsoft.com, and filtering the list of available newsgroups by using the word "security." Users often post their software update deployment experiences to these newsgroups.

You can also visit the Microsoft Security Web site for the latest security bulletin and security headlines at http://www.microsoft.com/technet/security/default.asp. Security bulletin updates and any issues that might affect your acceptance testing process are posted on this Web site.

Rollout Planning and Preparation

The production environment will need to be individually prepared for each new release. The tasks and activities that are required to prepare for rollout will depend on the release and the selected release mechanism. Common tasks include communicating information about the release to users and other personnel, training service desk and technical support staff, making backups of critical IT components, and configuring your release mechanism accordingly.

In preparation for the rollout, it will be necessary to provide advance communication to the users and inform support personnel of the deployment plan. Also, ensure that you have identified possible rollback procedures. Methods for uninstalling patches can vary from full uninstall support, manual uninstall steps, to no uninstall. Ensure sure there is a defined rollback plan should the deployment not match the success of the test environment.

Release Deployment

The process of moving the release into the production environment will depend on the type and nature of the release, as well as the selected release mechanism. In all cases, however, you will need to synchronize your test and production environment, and be prepared to track and monitor the progress of deployment.

Ideally, software updates should be released through a phased deployment. By following a phased deployment, you minimize the impact of any failures or adverse effects that could possibly be introduced by the initial distribution of a patch.

Change Review

As you deploy security patches within your organization, you will need to review and monitor your environment to ensure that patches have been applied successfully. The following sections discuss some of the steps you can take to verify successful patch installation. Considerations for stopping a deployment and for uninstalling a patch are also discussed.

Confirmation

The following actions will assist you with confirming the successful deployment of patches to your environment:

- **Use your vulnerability scanning reports to monitor the deployment of those software updates that they can detect.** Analyze scans of your environment after deployment to verify the software update is no longer reported as missing. You will need to determine when to best repeat the release management process to address computers that do not receive an update as expected.
- **Monitor computer health.** As a best practice, monitor a group of computers that represent your environment and look for any inconsistencies that might adversely impact users. You can also review event and system logs of computers in your environment for information regarding the success or failure of a security patch deployment. They might also include new error messages that could indicate a problem with a newly-applied patch.
- **Monitor service desk calls.** Look for new patterns of similar user calls. Common user complaints and system crashes could be related to the recent deployment of a security patch.

Rollback

As part of every patch deployment, you will need to define a rollback plan, should the deployment not match the success of the test environment.

The following are the main steps for the rollback and redeployment of patches:

- **Stop the current deployment.** Identify any steps necessary for deactivating release mechanisms used in your environment.
- **Identify and resolve any patch deployment issues.** Determine what is causing a patch deployment to fail. The order in which patches are applied, the release mechanism used, and flaws in the patch itself are all possible causes for a failed deployment.
- **Uninstall patches if necessary.** Patches that introduce instabilities into your production environment should be removed, if possible.
- **Reactivate release mechanisms.** After resolving patch issues, reactivate the appropriate release mechanism to redeploy patches.

Unfortunately, due to the fact that multiple installer technologies are used for patch installation, not all security patches can be uninstalled. Security bulletins issued by Microsoft will always indicate if a patch can or cannot be uninstalled. Because reverting computers to a previous state is not always possible, pay close attention to this detail before deploying a patch that cannot be uninstalled.

When a simple uninstall process is not available for a security patch, ensure the necessary provisions are in place for reverting your critical computers back to their original state in the unlikely event a security patch deployment causes a computer to fail. These provisions might include having spare computers and data backup mechanisms in place so a failed computer can be rebuilt quickly.

Post-Implementation Review

The post-implementation review should typically be conducted within one to four weeks of a release deployment to identify improvements that should be made to the security patch management process. A typical agenda for the review includes:

- Discussions on planned versus actual results.
- A discussion of the risks associated with the release.
- A discussion of lessons learned.

After the review, there are several potential follow-up actions, including determining whether additional changes are needed and distributing documentation about the lessons learned.

Techniques for a Security Release

Summary

List of Security Release Techniques	WU	SUS	SMS
Release Management Techniques			
Phased Deployment with SUS		Yes	
Phased Deployment with SMS			Yes
Change Review Techniques			
Confirming SUS and WU Deployments		Yes	
Confirming SMS Deployment			Yes

Release Management Techniques

The following techniques will assist you with implementing a phased approach for carrying out software update deployments.

Phased Deployment with SUS

Environment:	WU ☐	SUS ☑	SMS ☐
Purpose:	To control the release of patches by deploying them to different parts of your environment using a phased approach.		
Prerequisites:	A SUS infrastructure that consists of multiple servers.		
Scale:	Techniques scale to environments of all sizes.		

If your SUS environment consists of multiple servers, you can provide for a phased rollout by approving patches on the different SUS servers over time. You should first approve the update on the parent SUS server. After the update has been successfully deployed to clients supported by that server, you can synchronize the approvals list on the SUS child server that supports clients in the next phase of the rollout.

For example, an update that impacts all Windows XP clients is to be released to Seattle clients first and, after deployment is complete, to clients in Atlanta. In this example, the Atlanta SUS server is a child of the Seattle SUS server.

Upon approving an update on the Seattle SUS server, clients would begin to download and install the update. After deployment in Seattle is complete, and you have determined that the update is not adversely affecting your production environment, you could configure the Atlanta server to synchronize its approvals list with the list on the Seattle SUS server. As soon as the approvals lists have been synchronized successfully, updates approved on the Seattle server will be made available to clients supported by the Atlanta server.

By default, SUS servers are configured to synchronize their content directly from the Microsoft Windows Update servers, and updates need to be approved on each individual server. To reduce the amount of administrative overhead that is required by your phased deployment process, you can establish parent\child SUS server relationships that will help ensure that the proper update approvals take place throughout your environment. For more information, see the "Synchronizing Content With Another Server" section of the Software Update Services Deployment white paper: http://www.microsoft.com/windows2000/windowsupdate/sus/susdeployment.asp.

Note: Another technique for phasing deployments involves setting the installation schedule of the Automatic Updates client differently for groups of clients.

Phased Deployment with SMS

Environment:		WU ☐	SUS ☐	SMS ☑
Purpose:	To control the release of patches by deploying them to different parts of your environment using a phased approach.			
Prerequisites:	A working and healthy SMS hierarchy.			
Scale:	Techniques scale to environments of all sizes.			

Depending on the size of your SMS hierarchy, you can create Collections based on known subnets and distribute the software to a subset of the population. This phased operation allows you to monitor the distribution more effectively.

To perform a phased deployment:

1. Determine your subset criteria. Will you distribute by subnet? By operating system? By Microsoft Active Directory® site or container participation? By department? Your environment and business can help determine this for you.
2. Create Collections, based on an SMS query that sorts the SMS-managed clients. For example, sort by servers in one SMS Collection, workstations by organizational unit, SMS Site, or Active Directory site in another.
3. Create the Package.
4. Create the Advertisement.
5. Assign the Advertisement to your first Collection.
6. After you have identified that the deployment was successful to the first group, assign the Advertisement to your second Collection, and so on.

Change Review Techniques

The following techniques will assist you with monitoring progress and rolling back software update deployments.

Confirming SUS and WU Deployments

Environment:	WU ☑ SUS ☑ SMS ☐
Purpose:	To determine if the computers in your environment have successfully installed patches deployed using SUS or WU.
Prerequisites:	Must have implemented MBSA vulnerability scans, as described in Part II, Chapter 2, "Setup."
Scale:	Reviewing event logs in large environments usually needs to be limited to critical computers. IIS reporting works efficiently in environments of all sizes.

Analyze Vulnerability Scans, Event Logs, and Registry

The following actions will assist you with confirming the successful deployment of software updates to your environment:

- **Use MBSA to monitor the deployment of those software updates that it can detect.** Analyze an MBSA scan of your environment after a patch has been deployed to verify the update is no longer displayed as missing in the MBSA report. You will need to develop a plan for addressing computers that do not receive an update as expected. In most cases, you can wait until computers resynchronize with SUS servers in your environment, but depending on the severity and the level of exposure your environment has to a given security vulnerability, more drastic actions might need to be taken.
- **Check computer registry.** Many software updates log information to the following registry location:

  HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Updates

  Update information is arranged by product, and includes a description of the update, who installed the update, the update installation date, and uninstall command, if applicable.

  The Additional Information section of a security bulletin provides the registry entries that are made by the security patch installation package for any applicable operating systems.
- **Check the event log for any SUS-related errors on Automatic Updates (AU) clients.** The complete list of possible events for the AU client is in Appendix C, "Client Status Logging" of the Software Update Services Deployment white paper: http://www.microsoft.com/windows2000/windowsupdate/sus/susdeployment.asp.
- **Review the IIS logs on the SUS server in your environment.** Verify the http return codes (for example, "HTTP 404 – File not Found") for each entry to establish if computers are experiencing any download failures.
- **Look for errors in the "Windows update.log" file on critical computers in your environment.** This file contains a log of all activities performed by the AU client. The file is located in the root of the system folder on each client computer (for example, C:\Windows).

- **Verify the state of the AU client in the registry.** The **AUState** value will indicate if the installation of any patches is still pending. This value is stored in the following location:

  HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\WindowsUpdate\Auto Update

  The value for **AUState** must be set to 2 for the client to look for newly-approved updates. If **AUState** is set to 5, the client is pending installation of downloaded updates and will not run a new detection until these updates have been installed. The client will reset its **AUState** to 2 after completing installation.
- **Watch for system instabilities.** As a best practice, monitor a group of computers that represent your environment to look for any system outages that might adversely impact users. Monitor your service desk calls for patterns of common user complaints and system crashes that could be related to the recent deployment of a security patch.
- **Check for uninstall folder.** Uninstall information for many patches is stored in a hidden folder under the system folder (for example, C:\Windows\$NtUninstallQ331320$). The presence of these folders on a system is a good indication that a patch has been successfully installed.

Configure IIS Reporting

Automatic Updates (AU) clients can be configured so they return status information to the SUS server or any other IIS 5.0 server that has logging enabled. The AU client status information is displayed as raw IIS log data within the IIS logs. For detailed information on how to correctly interpret log information, and on how to optimize IIS for AU client logging, see Appendix C, "Client Status Logging" in the Software Update Services Deployment white paper: http://www.microsoft.com/windows2000/windowsupdate/sus/susdeployment.asp.

The AU client reports its status to the specified server on these instances:

- During Self-update: self-update pending
- After Self-update: self-update success/failure
- During Detection: initialization success/failure
- After Detection: detection success, detection failure
- After Download: download success/declined/failure
- After Installation: installation success/declined/failure

Reviewing these logs provides you with an accurate and efficient method for monitoring the installation of a security patch as it is released into your environment.

Note: SUSserver.com provides a SUS reporting utility to build a report based on the IIS log file of a SUS server. For more information, see SUS Reporting Utility: http://www.susserver.com/Software/SUSreporting/.

SUSserver.com is not affiliated with or endorsed by Microsoft.

Uninstalling an Update

If you determine that you need to uninstall a security patch, you can often do so by using the **Add/Remove Programs** item in Control Panel. Uninstalling a patch could be a daunting task in most environments. However, it is often possible to remove a patch by running its uninstall command. The following is an example of a typical uninstall command:

```
C:\WINNT\$NtUninstallQ331953$\spuninst\spuninst.exe
```

You can include uninstall commands in a login script or computer startup scripts and assign it using Active Directory Group Policy to automate the removal of many software updates.

Note: This particular command would uninstall the patch related to security bulletin MS03-010. The Knowledge Base article number that contains specific information about the security patch is referenced as part of the name of the uninstall folders. The previous example references KB article 331953.

Confirming SMS Deployment

Environment:	WU ☐	SUS ☐	SMS ☑
Purpose:	To determine if the computers in your environment have successfully installed patches deployed using SMS.		
Prerequisites:	Healthy software distribution and status systems. The Web Reporting tool installed on a server running IIS.		
Scale:	Techniques scale to environments of all sizes.		

Systems Management Server provides the following methods for reviewing the success or failure of a patch deployment:

- SMS Administrator Console Advertisement Status
- Web Reporting Tool
- Advertisement Status Viewer
- SMS Log Files

SMS Administrator Console Advertisement Status

In the SMS Administrator console, Advertisement Status contains a status summarizer for each advertisement that is distributed by SMS, and a summary of advertisement statistics across all sites in the hierarchy. This information allows you to proactively monitor the patch deployment process. The Status System includes several console items describing the status of software distributions:

- Package Status Summarizer
- Advertisement Status Summarizer
- Package Detailed Information
- Advertisement Detailed Information
- Informational, warning, and error messages

Web Reporting Tool

The Microsoft Web Reporting tool comes with the SUS Feature Pack for SMS. Installed as an IIS site, the Web Reporting tool provides a graphical, Web view of the SMS data. You can view the software update reports provided by the Web Reports Add-In for Software Updates in the list under the **Software Updates** node in the Web Reports tool.

The Web Reports Add-In for Software Updates contains several pre-configured reports that you can use to view software update–specific information. In addition to using the preconfigured reports, you can also use SQL Server views and the documented inventory schema to create custom software update inventory reports that are tailored to the needs of your enterprise.

For more information about the Web Reporting Tool, see the Microsoft WebCast, "Microsoft Systems Management Server Reporting using the SMS Web Reporting Tool": http://support.microsoft.com/?kbid=325071.

Advertisement Status Viewer

You can use the Advertisement Status Viewer tool to view the comprehensive status of an advertisement on individual SMS clients or collections over various user-defined intervals.

The Advertisement Status Viewer:

- Displays the status of a given advertisement on all of the clients that should receive the advertisement—that is, all of the clients to which the advertisement is targeted.
- Shows you which clients have not received the advertisement. This can help with laptops, and so forth. (Knowing which clients haven't received the advertisement is difficult to tell from the status messages, because this condition exists when the client has not yet reported a status message about the advertisement.)
- Displays **% complete** statistics, such as the percentage of clients that have successfully run the advertisement, the percentage that have not yet received it, and so forth.
- Handles situations in which an advertisement is targeted to a collection and all of its subcollections. In this case, recursive queries are issued to determine the total client set in the collection.
- For a given collection, displays the status of all of the advertisements that each computer in the collection should receive. The status is not limited only to the advertisements targeted to that collection, but also includes those targeted to any collection to which each of those clients belong.
- Displays the raw text of the status messages. In addition to raw message text, it also displays text of all WMI classes associated with a given status message.
- Can print and save status.
- Can copy status into other applications.

The Advertisement Status Viewer is part of the Systems Management Server 2.0 Service Pack Support Tools:
http://www.microsoft.com/smserver/downloads/20/tools/spsupport.

SMS Log Files

In addition to sifting through the information provided by SMS through the site services, you can review log files that reside on both the client and the server. Reviewing both the site server information and the client-side log files gives you the complete picture for troubleshooting problems effectively. Sometimes reviewing information in both locations is the only way to resolve a software delivery issue.

For more information about SMS Server and Client Log Files, see the SMS Product Documentation, Appendix D - System Flows: http://www.microsoft.com/technet/prodtechnol/sms/proddocs/smsadm/appendixes/smsadad.asp.

Note: The default log file setting for the SMS client is 100 KB. In some cases, this size of log file may not provide enough information to diagnose a problem.

You can change the file size on the client by modifying the appropriate registry key:

HKEY_LOCAL_MACHINE\Software\Microsoft\SMS\Client\Configuration\Client Properties

Value Name: Log File Size for Debugging

Data Type: DWORD

Value: 400 (hexadecimal) for a 1-MB maximum log file size

The SMS Advertised Programs Client Agent uses Status Messages numbers 10000 to 10021. You employ these Status Message numbers to monitor the Software Distribution process and also to build your own Status Message Queries. The Status Message Queries that come installed with SMS are based on these Message ID numbers.

To view the full range of status messages that SMS reports, see SMS Resource Kit, Chapter 26, "Status Message": http://www.microsoft.com/technet/prodtechnol/sms/reskit/sms2res/part8/smc26.asp.

Rolling Back an SMS Deployment

To roll back a deployment using SMS:

1. Stop the deployment of the active package.
2. Identify issues and resolve.
3. Uninstall the update.
4. Re-advertise the package.

If a deployment is unsuccessful, you must have a plan in place to stop the rollout, uninstall failed updates, and then redeploy them. SMS provides this capability through the following methods.

Stopping the Deployment of an Active Package

From the packages node of the SMS administrator console, select the program that you want to stop temporarily. On the **Program Properties Advanced** tab, select the **Disable this program on computers where it is advertised** check box. Doing so will stop or pause advertisements based on the program even if they are assigned as mandatory and set to begin at a specific time or on a schedule.

When you select the **Disable this program on computers where it is advertised** check box, the advertisement for the program will stop or pause until you clear the check box. This method of stopping the deployment of an active package is most commonly used when you need to replace existing source files on your distribution point servers. While the deployment of an active package is stopped or paused, you can test the package program or replace files. Selecting this check box leaves the source files intact and does not delete them or the folder in which they are contained.

Re-advertising a Package

You can easily accomplish resending the same package by adding an additional schedule to the existing advertisement. This procedure will force the client to reinstall the package, even if it has received the package and run it before.

If you have deleted the advertisement and need to resend the package to the same clients, you can create a new advertisement and add *two* schedules. The first schedule in the advertisement shouldn't run because the client compares the package ID and knows it already ran the advertisement. The second schedule in the advertisement will be the "true" schedule and the client will recognize that the package ID is new and rerun the advertisement.

You can keep rerunning the same advertisement over and over by adding another schedule to the advertisement. This is a great solution when you're doing test jobs on your lab computers and need to quickly resend the package. This procedure can help you adequately test the patches before you are ready to deploy them in your production environment.

Uninstalling an Update

Some of the updates released by Microsoft provide an uninstall path, whereas others do not. You will have to determine which patches can be uninstalled by reviewing the technical details of the security bulletin. To uninstall a patch using SMS, you just need to know the uninstall command. Microsoft updates generally place a reference into the registry, including the uninstall command when an uninstall is possible. You can create an SMS package that retrieves this information from the registry, and then acts upon it. You distribute this package the same way you would deploy the release, by using the SMS Advertisement procedures.

SMS includes a program called SMS Installer. SMS Installer is a package-creation tool. You can use SMS Installer to generate the package to uninstall a patch that is listed in the **Add/Remove Programs** registry key. In the **Add/Remove Programs** registry key, each update is listed by its Knowledge Base article number, for example: 331953. Each update generally utilizes a special uninstall command that also references the Knowledge Base article number. The general uninstall command example is as follows:

```
C:\WINNT\$NtUninstallQ331953$\spuninst\spuninst.exe
```

Using SMS Installer to retrieve this information and initiating the uninstall command line is relatively simple. SMS Installer Script.ipf, an SMS Installer script in the SMS Tools and Templates that accompany this guide, shows how the process works.

For more information about SMS Installer, see the SMS books list: http://www.microsoft.com/smserver/techinfo/books.

5

Enforcing Security Policy

Summary

Deploying a security patch does not eliminate the risk associated with the vulnerability.

Many organizations have some level of decentralized administration (such as multiple groups that have administrative rights or end-users who have administrative rights on their computers), unrestricted or non-existent baseline computer installation standards, or unmanaged computers (such as lab computers or "servers under desks") within their environment.

There may be valid business reasons for any of these choices, or the IT environment may not have disciplined administrative controls in place. In either case, vulnerability management is an important activity that is closely related to security patch management.

Security vulnerabilities will arise in these environments even after the security release process has effectively deployed a security patch. Vulnerabilities can easily recur when new computers are installed or reinstalled, software is installed, configurations are changed, or unmanaged computers aren't addressed by a security release.

The entire activity and cost of security patch management is wasted without a comprehensive approach to dealing with vulnerabilities that recur in the environment. Recurring vulnerabilities are potentially more dangerous than when they were first announced—the probability of an exploit existing and attack tools identifying and exploiting the vulnerability is much higher.

Note: For more information about how the Microsoft® Operations and Technology Group manages vulnerabilities, see Managing Computer Vulnerabilities at Microsoft: http://www.microsoft.com/technet/itsolutions/msit/security/mscomvul.asp.

Security Policy – Security Standards

To realistically enforce a security policy, the policy must exist and be communicated to the people who are expected to abide by it. An effective security policy should identify the minimum security standard for computers (based on your organization's risk tolerance), which may include:

- Installation standards, describing supported installation locations and methods.
- Network and domain standards, indicating how names and TCP/IP information is assigned and which domains computers should join.
- Operating system security options and policy settings, including reducing open ports based on required services.
- Any standards that describe the use of encrypted file systems.
- Minimum service pack and security patch compliance, updated with each security release.
- Anti-virus software compliance.
- Application security configuration settings, such as macro file protection and security zones.
- Administrative account standards, such as renaming or disabling accounts and setting up decoy accounts.
- Strong password standards.

A security standard may differ for each computer category (such as workstations, servers, and computers that connect remotely) and should evolve over time.

Your security standards and software baselines should be closely related. Security standards might provide input into your baselines, or new baselines can provide input into your security standards.

A security policy violation suggests a vulnerability that should be eliminated from the environment. The vulnerability could be a computer that lacks various software updates, is misconfigured, or a user failing to use strong passwords.

The security policy may provide guidelines for each type of policy compliance violation; this can then be used to determine the severity of an incidence of a specific vulnerability.

Note: Microsoft Baseline Security Analyzer (MBSA) scans for security-related software updates and common security misconfigurations. It does not scan for conformance with all elements of an organization's security policy. Based on the specifics of your security policy, you may want to add additional vulnerabilities to your vulnerability scanning reports through scripting and other tools.

After a security policy that defines computer security standards is in place, apply the strategies presented in the following section to address any discovered violations/vulnerabilities through a series of escalations.

Enforcement Strategies

A security vulnerability, such as a weak password or a computer without a required security patch, that violates security policy should be treated like a standard issue through your service desk.

Enforcing security policy can be complicated by distributed administration and vulnerable assets that are not centrally managed. The person or group who is responsible for administering the asset and resolving the vulnerability may be unknown or hard to find, could physically reside within another department in the organization, or may not have the necessary skills to resolve the vulnerability on their own.

Accordingly, there are several practices that become necessary and helpful when it comes to enforcing security policy:

- Security policies, enforcement timelines, and approaches should have the support of upper management across departments.
- Techniques and tools for determining ownership and administrative information on an unmanaged computer should be available to specific service desk technicians.
- Service desk technicians should be trained on mitigating each of the vulnerabilities that violate security policy, so that they can help others address the vulnerabilities that are assigned to them.

Note: There are several scripts available on the TechNet Script Center to help identify computer information remotely, if you have administrative access:

- Retrieve System Information: http://www.microsoft.com/technet/scriptcenter/compmgmt/scrcm42.asp.
- Retrieve Operating System Properties: http://www.microsoft.com/technet/scriptcenter/compmgmt/scrcm40.asp.
- Identifying the User Logged on to a Remote Computer: http://www.microsoft.com/technet/scriptcenter/user/scrug59.asp.
- Returning Attribute Values for a Local User Account: http://www.microsoft.com/technet/scriptcenter/user/scrug116.asp.

If you don't have administrative access to a computer, your options for identifying information about it are greatly reduced. However, there are port-scanning utilities and other security tools that can be valuable for this purpose.

For more information, see Part I, Appendix A, "Third-Party Tools and Resources."

Approaches and Timelines

The security policy should define which vulnerabilities are prohibited in the environment, and describe the enforcement approach and related timelines.

The nature of a vulnerability, such as the risk of exploitation and the cost of recovery, should help determine how aggressive the response should be. Some vulnerabilities should be resolved within 24 to 48 hours, whereas others may reasonably take one to two weeks to resolve.

If the vulnerability will be resolved with a software update, you can attempt some basic approaches, including:

- Asking the asset owner to visit Microsoft Windows® Update or an internal site to install the appropriate software updates.
- Forcing the software update upon the computer through your patch distribution infrastructure.

If attempts to resolve the vulnerability within the required timeline are unsuccessful, you may choose to employ more aggressive tactics, including:

- Escalating the issue within the violator's organization.
- Disabling the primary account that is used to access the computer.
- Removing the computer from the network by physically disconnecting it or configuring network hardware to have the same effect.

These last two tactics will typically result in a call to the service desk, where the unresolved vulnerability can be discussed and an acceptable resolution determined. Be sure to have executive support for the security policy before escalating security violations and attempting these techniques.

Note: The TechNet Script Center includes scripts for disabling and enabling user accounts:
http://www.microsoft.com/technet/scriptcenter.

- Disable a User Account:
  http://www.microsoft.com/technet/scriptcenter/user/scrug20.asp.
- Enable a User Account:
  http://www.microsoft.com/technet/scriptcenter/user/scrug25.asp.

For an introduction to scripting, see System Administration Scripting in the Windows Environment:
http://support.microsoft.com/default.aspx?kbid=325946.

6

Emergency Security Response

Summary

Even with the best patch management process, your technology environment can still be successfully attacked. Not all vulnerabilities are resolved by the application of software updates, but may be related to weak computer security configurations.

Alternatively, a software vulnerability could be exploited before a software update is available, or even before it has been publicly reported—otherwise known as a "zero day" attack. Perhaps a vulnerability that has recurred in the environment has been exploited before being addressed.

Regardless, it is not necessary to understand why you are vulnerable to realize that an emergency security response may be necessary. To deal with the emergency effectively, you need to have an incident response plan in place. This chapter identifies key ways to prepare for an emergency, provides several ideas for an incident response plan, and gives prescriptive measures and ideas on how to minimize impact and take control during an emergency situation.

Preparing for an Emergency – Contingency Planning

Contingency planning for an emergency security response is how your organization should prepare before an attack occurs—ensuring a smooth, coordinated response during a time when every second can limit the damage caused by an attack.

There are three main areas that should be prepared in advance of an emergency security response:

- Regular auditing and intrusion detection tools and practices.
- An identified Incident Response team with a formal Incident Response plan, including isolation and containment techniques that can be implemented quickly across your entire environment.
- Established post-incident actions and a review process to learn from the attack.

Consider the following types of attacks, which may involve different people or different approaches throughout the incident response plan:

- Virus or worm attacks.
- Distributed denial of service (DDOS) attacks.
- Unauthorized network intrusions.
- Internal network abuse.

Some additional best practices that will be of benefit in preparation for an attack:

- Maintain secure operating system images (baselines) that can be quickly restored to a given hardware platform to quickly revert the system to a known good operating system.
- Ensure that your organization is trained on all of the techniques they'll require during an attack.
- Ensure your end-users are trained on security policies and acceptable use policies.
- Maintain a prioritized list of all key information assets that should be protected first in an emergency.
- Follow all of the recommended setup activities for security patch management in this guide, including subscribing to security notifications and establishing ongoing vulnerability scanning reports. These are discussed in Part II, Chapter 2, "Setup."
- Establish and maintain disaster recovery information for all systems. Consider investing in failover assets (possibly at secondary locations) and procedures for critical systems.
- Keep all software updates on CD for use when Internet access may be impacted.

Note: Microsoft® Windows® Update provides an online catalog of software updates that can be downloaded. In Windows Update, under **Other Options**, click **Personalize Windows Update**, and then select the **Display the link to the Windows Update Catalog under See Also** check box.

Windows Update:
http://www.windowsupdate.com.

Office Update also includes the Office Download Center for downloading Office software updates:
http://office.microsoft.com/downloads/.

Avoiding Attacks with Strong Security Configurations

In addition to security patch management, there are several best practices that can help you avoid attacks through secure computer configurations. For example, you can rename built-in or default accounts and groups, because these accounts are commonly attacked.

There are several comprehensive resources available to assist with securing various Microsoft products.

Securing Windows 2000 Server:
http://go.microsoft.com/fwlink/?LinkId=14837.

Windows Server 2003 Security Guide:
http://go.microsoft.com/fwlink/?LinkId=14845.

Windows XP Security Guide:
http://go.microsoft.com/fwlink/?LinkId=14839.

Threats and Countermeasures: Security Settings in Windows Server 2003 and Windows XP:
http://go.microsoft.com/fwlink/?LinkId=15159.

Security Operations for Exchange 2000 Server:
http://www.microsoft.com/technet/security/prodtech/mailexch/opsguide.

Securing Wireless LANs:
http://go.microsoft.com/fwlink/?LinkId=14843.

Microsoft Internet Security and Acceleration (ISA) Server, a multilayer enterprise firewall and Web cache, can be configured to help stop worms and viruses before they enter your network. For more information, see Preventative Measures:
http://www.microsoft.com/isaserver/techinfo/prevent.

Detecting Intrusions

Some attacks have been easy to identify; for example, the Melissa and Anna Kournikova e-mail viruses made their presence clearly known. Code Red defaced Web servers. Others, like SQL Slammer, didn't leave any obvious signs other than an increase in network traffic.

There are many resources on the Web that provide early indicators of unusual activity:

- CERT Current Activity (includes checklists and guidelines for incident response): http://www.cert.org/current/.
- Internet Storm Center: http://isc.incidents.org.
- Computer Incident Advisory Capability (CIAC): http://www.ciac.org/ciac/.

Network Monitoring

A combination of tools and practices are required to detect intrusions inside the network or on the perimeter. The following are some activities and identifiers to look for while performing regular network monitoring:

- **Look for sudden increases in overall traffic.** An increase in traffic can be explainable, for example if your Web site was mentioned on a popular news site. However, a sudden and unexpected increase in overall traffic could indicate a problem. Baseline router and firewall traffic and check regularly for increases above the baseline.
- **A sudden jump in the number of bad or malformed packets.** Some routers collect packet-level statistics; you can also use a software network scanner to track bad packets.
- **Large numbers of packets caught by your router or firewall's egress filters.** Egress filters prevent spoofed packets from leaving your network, so if your filter is catching a large number of packets, you need to identify their source, because the computers on your network could have been compromised. Check your network hardware documentation for more information on egress filters.

Note: Network Monitor is a Microsoft tool that comes with Microsoft Systems Management Server (SMS) that can be used to capture network traffic. (A limited version is also included with Windows 2000.) For more information on Network Monitor, see HOW TO: Use Network Monitor to Capture Network Traffic: http://support.microsoft.com/?kbid=812953.

Also see Frequently Asked Questions About Network Monitor: http://support.microsoft.com/?kbid=294818.

ISA Server also includes several capabilities to assist with detecting and responding to intrusions, described in Security with Internet Security and Acceleration Server 2000: http://www.microsoft.com/isaserver/techinfo/planning/firewallsecuritywp.asp.

Monitoring Configurations

There are several areas to check on computers and within domains:

- **Unauthorized services.** Check the list of running services on the server or workstation. Verify that each service is required by comparing with similar computers. Many attacks use service names that seem legitimate—baseline running services on new computers and periodically check for changes.
- **Hidden files or new directories.** Locate any hidden files located on C:\, in the C:\Winnt or C:\Winnt\System32 directories or elsewhere in the executable path. Look for any newly-created directories that may contain attack code.
- **Appearance of new or unknown user accounts.** The appearance of new user accounts, especially ones assigned to privileged groups, can indicate a successful attack or privilege escalation. Watch for accounts that don't match your organization's naming conventions.

Note: The TechNet Script Center includes several scripts for remotely listing services, folders, and domain user accounts.

Determine Services Running in All Processes:
http://www.microsoft.com/technet/scriptcenter/services/ScrSvc04.asp.

Enumerate all the Folders on a Computer:
http://www.microsoft.com/technet/scriptcenter/filefolder/ScrFF25.asp.

Enumerating all the User Accounts in an NT 4.0 Domain:
http://www.microsoft.com/technet/ScriptCenter/user/ScrUG120.asp.

Auditing Event Logs

Event logs are also a source of useful intrusion information. Check application, system, security, and IIS logs and watch for the following:

- **Unscheduled restarts of server computers.** These may indicate that they've been compromised. Monitor server event logs for failed logons and other security-related events.
- **Unscheduled restarts or unavailability of server applications.** Take notice of any server applications, such as Microsoft Internet Information Services (IIS) or Microsoft SQL Server™, where application services restart unexpectedly, or become unresponsive.
- **Failed file access audit events.** Many attacks will attempt to access system files that may be protected by NTFS permissions. If your site has auditing enabled, you may find a pattern of failed access attempts pointing to an attack in progress.

Note: For more information about auditing in Windows, see:

HOW TO: Enable and Apply Security Auditing in Windows 2000:
http://support.microsoft.com/?kbid=300549.

Best Practices for Auditing Windows 2000:
http://www.microsoft.com/windows2000/en/server/help/sag_SEconceptsImpAudBP.htm?id=420

A regular audit of your logging policies should be addressed because log files can roll over in many applications. You should decide if you need to generate processes for backing up log files periodically—before they are overwritten.

There are several tools available to help with monitoring event logs, including:

- **Log Parser 2.0.** A powerful, versatile tool that you can use to extract information from files of almost any format by using SQL-like queries.
  Log Parser 2.0:
  http://www.microsoft.com/windows2000/downloads/tools/logparser.
- **EventCombMT.** A multi-threaded tool that will search event logs from many servers at the same time, and which is included in the Windows Server 2003 Resource Kit.
  Windows Server 2003 Resource Kit:
  http://go.microsoft.com/fwlink/?LinkId=4544.
- **Command-line tools.** Tools such as EventQuery.vbs and EventTriggers.exe are available with Windows XP and Windows Server 2003 to help manage events.
  Managing event logs from the Command Line:
  http://www.microsoft.com/technet/prodtechnol/winxppro/proddocs/event_commandline.asp.
- **Microsoft Operations Manager.** Provides enterprise-class operations management including comprehensive event management, proactive monitoring and alerting, reporting, and trend analysis.
  Microsoft Operations Manager:
  http://www.microsoft.com/mom.
- **Additional resources.** For detailed explanations, recommended actions, and links to additional support and resources for various events and error messages, see:
  Events and Errors Message Center:
  http://www.microsoft.com/technet/support/eventserrors.asp.

Note: For additional information on auditing and intrusion detection, including details on Event IDs to watch for, see Auditing and Intrusion Detection:
http://www.microsoft.com/technet/security/prodtech/windows/secwin2k/09detect.asp.

There are also several commercial intrusion detection systems that are, which use known attack signatures to help with intrusion detection activities. For example, intrusion detection systems can check for Code Red by scanning the IIS log file for a distinctive GET request for default.ida. Network packets are also scanned for known attack signatures.

Several intrusion detection systems are listed in Part I, Appendix A, "Third-Party Tools and Resources."

Incident Response Plan

If you are faced with an active attack, there are measures that you can take to reduce the impact of the intrusion.

Good incident response plans identify the actions that will be performed during an attack and ensure that all roles and responsibilities are understood throughout the organization, as well as the triggers that initiate these actions.

Note: For more information on Responding to Incidents, see:
http://www.microsoft.com/technet/security/prodtech/windows/secwin2k/10respnd.asp.

CERT has a Handbook for Computer Security Incident Response Teams:
http://www.cert.org/archive/pdf/csirt-handbook.pdf.

For sample incident handling documents, see the incident handling forms hosted by SANS:
http://www.sans.org/score/.

Evaluation

The first step in the emergency response scenario is to identify and define the emergency. If you are in an emergency situation, the attack requires immediate attention. How vulnerable are all of your systems? You need to be able to immediately prioritize the resources needed to fight the emergency based on asset valuation.

When your intrusion detection system or other indicators tell you that you're under attack, as part of your evaluation, you need to:

- **Identify the nature of the attack.** Is the attack a DDoS, or an attack targeted just at you? Is someone trying to shut down your network altogether, or attempting to infiltrate individual computers?
- **Localize the source.** Use your firewall and audit logs to attempt to identify where the attack originated. This will help you identify whether the attack is coming from a compromised host on your own network or from the outside world.

Notification and Escalation

Proper communication is critical to managing an attack. Depending on the type of attack, determine exactly who needs to know that an attack is underway. During a targeted attack, don't tip off the attacker with company-wide communications—keep attack communications contained to the people who have a need to know.

With a virus or worm, company-wide communication that reaches all employees onsite and remote may be appropriate. Keep in mind that communication technologies may also be under attack—have a contingency plan in the event any of your usual methods are unavailable.

Be sure to communicate appropriately. Make the communication to the point, informational, and constructed to alleviate any panic. If you have specific company guidelines for communications, make sure to follow them so that the recipients trust the message. These guidelines may need to be revised to accommodate emergency situations.

It is very important that all those who are involved in an incident response communicate effectively. Doing so will help ensure that decisions are made without duplicating effort, and that no steps of the process are missed. The incident response team should be the focal point for all communications.

Contact Your Technology Vendors

If a Microsoft product is involved in the attack, notify Microsoft Product Support Services at (866) PC SAFETY for free virus and security patch–related support in the United States and Canada. For other locations, see Microsoft Product Support Services worldwide:
http://support.microsoft.com/common/international.aspx.

If you have Microsoft Premier Support, contact your Technical Account Manager using their contact information.

Microsoft has security and incident response experts that can help you understand an attack and assist you throughout the incident response process.

Contact Law Enforcement Agencies

Intrusions can be a criminal event. Consider notifying the appropriate law enforcement agency if you are under attack, and understand and comply with your local jurisdictional requirements for involving the authorities and informing those that may be affected by the intrusion.

Many government agencies have significant experience assisting organizations with Internet intrusions and subsequent prosecution—likely more experience that your organization will have. Reduce your risk in an emergency situation by involving them!

Note: In the United States, there are several agencies that you can contact when you are under attack. The United States Department of Justice provides information on How to Report Internet-Related Crime:
http://www.usdoj.gov/criminal/cybercrime/reporting.htm.

Contact Legal Counsel

It is a good idea to contact your organization's legal counsel at this point to collaboratively determine if legal prosecution is a possibility after the event, depending on the nature of the attack. If legal prosecution is an option, throughout the process your organization should maintain a log of the business impact of the attack (damages), and take additional steps to protect the evidence, such as:

- Keep backup copies of any logs you generate on read-only media, and take detailed notes so that you have a good evidential record of what happened and when. Include checksums of all data collected on the same read-only media to prove that it wasn't tampered with.
- Save the running system state (services, ports open, user accounts, memory maps, and so on) and create forensic image of the suspect drive. There are forensics tools available to help with this, listed in Part I, Appendix A, "Third-Party Tools and Resources."
- If protecting evidence is critical and a backup computer (and data) is available, don’t attempt to change or fix the affected computer. Power it off and introduce the backup computer after ensuring that it does not have the vulnerabilities that would make it susceptible to attack. Just like a crime scene, the more you do to an affected computer the greater the chance of destroying evidence.

Isolate and Contain

Although uptime is very important in most environments, keeping computers available during an attack may result in more damage. Balance the impact of an ongoing attack with the impact of an appropriate defense.

Attacks that destroy, manipulate, or divulge sensitive data, or require a full computer reinstallation to recover from, may merit taking computers offline to protect them. Less intrusive attacks, like some denial of service attacks, may not require a response this severe. In most cases, the source of an attack should be removed from the network regardless of the type of attack to isolate and minimize proliferation.

When you take extreme measures that impact the business, you should track them closely so that they can be successfully removed after the incident is resolved.

The following are several activities that you may choose to perform to isolate and contain an attack:

- **Disable access points.** Determine which access point(s) the attacker used and implement measures to prevent ongoing access. Such measures may include disabling a modem, shutting down virtual private network (VPN) and remote access servers, adding access control entries to a router or firewall, or physically disconnecting network equipment.
- **Protect classified, sensitive and proprietary data.** As part of planning for incident response, you should clearly define which assets contain sensitive information that needs to be protected. Depending on the nature of the attack, you may choose to shut down these computers or turn off specific services.
- **Protect software against attack.** This includes protecting against loss or alteration of system files. Damage to software can result in costly downtime.
- **Block the attack**. If an attack or attempted attack is coming from outside, block access to your network from that IP address. Some attacks change the range of the source IP address, so you may need to analyze the traffic and block specific ports.

If you're the target of a distributed denial of service (DDoS) attack, you may want to work with your ISP on a coordinated response.

There are also specific items you may want to turn off or disable during an attack. Some of these are:

- **Remove attack-source computers**. If you've identified specific computers that have been compromised and are involved in the attack, you may want to pull them from the network until you can disinfect them and return them to service.
- **File shares.** If the attack uses the context of the user to gain access to files and data for destruction, set all file shares to read-only to allow most work to continue but prevent attacks from causing further damage.
- **Virtual private network or remote access.** To protect your remote users from the attack, you may choose to disable the ability for remote users to connect to the company network. The attack could spread to those users who are connecting, or the attack could be coming from a remote user.
- **Internet connection.** Shut down connection to the outside world. Not only could this halt the attack should it be coming from outside of the company, but it can also contain the attack before you infect other locations.
- **Internal connections.** Try to contain the attack by disabling connections to other company offices or geographic locations.
- **Stop services and applications.** Certain vulnerabilities depend on specific services running to propagate themselves. Shut down services that could proliferate the attack.
- **Shut down or block ports.** Block ports at the firewall. Vulnerabilities that attack from the outside generally depend on specific ports or port numbers to be open. Review the documentation from your networking hardware vendor for more information.
- **Change passwords for elevated privilege accounts.** Various vulnerabilities attempt to guess the password for specific accounts. Administrator and Guest accounts and accounts that run with high privilege are favorite points of attack. Change these passwords immediately when confronted with an attack.

Note: For more information on using IPSec to block ports and secure a server, see Using IPSec to Lock Down a Server: http://www.microsoft.com/technet/itsolutions/network/maintain/security/ipsecld.asp.

For scripts that can start and stop services remotely, see the TechNet Script Center for Services: http://www.microsoft.com/technet/scriptcenter/services.

For scripts that can remotely change passwords and lock accounts, see the TechNet Script Center for Users and Groups: http://www.microsoft.com/technet/scriptcenter/user.

Analyze and Respond

To be able to recover effectively from an attack, you need to determine how seriously your environment has been compromised. This will help identify how to contain and minimize the risk, how to recover, with whom you should communicate the incident, and whether to seek legal redress. In general you should attempt to:

- Determine the nature of the attack, which may be different than your initial assessment suggests.
- Determine the point of origin of the attack.
- Try to determine an attack signature so you can recognize when new computers are being attacked.
- Determine the intent of the attack. Was the attack specifically directed at your organization to acquire specific information, or was it a random attack?
- Identify which computers have been compromised.
- Identify the files that have been accessed and determine the sensitivity of those files.

For more information about security tools that are available to help with computer forensics and analysis, see Part I, Appendix A, "Third-Party Tools and Resources."

Remediation

If computers need to be recovered from an attack, it is always best to rebuild a fresh system. Consider rebuilding a fresh system with new hard disks using your up-to-date, secure baseline. Ensure that you change any local passwords and address the vulnerability that was exploited during the attack. You should also change administrative and service account passwords elsewhere in your environment.

If Microsoft or your virus vendor provides a recommended way of cleaning a computer from a virus or worm that doesn't require a full reinstallation, this option may be preferable because of the time and cost saved.

However, there is always a chance that an attacker opened several back doors after your computer was compromised during an attack (for example, several viruses leave back doors for future exploits). When a computer has been compromised by an attack of any kind, the safest way to return it to service is to reinstall the operating system, reload applications from a known good backup or baseline that applies an up-to-date security policy, and ensure the exploited vulnerability has been addressed.

Even though it might be tempting to address the compromise quickly and get the computer back on the network, it's risky to do so, because it is impossible to determine what back doors or changes to the computer attackers have left.

Note: CERT maintains a list of Steps for Recovering from a System Compromise: http://www.cert.org/tech_tips/root_compromise.html.

Accelerated Release Management

When fixing problems across the environment in response to an attack, use your current change and release management processes as a guide, performing only those steps that are required to quickly and effectively respond to the attack.

If an attack can be resolved with the application of a software update or countermeasure, determine how to install it throughout the organization quickly. The risk of the vulnerability being exploited during an attack is significantly higher than normal, so you may decide to perform only basic testing during an emergency.

Take all of the company's technology assets into account. During an attack, not all assets may be connected to the network—for example, remote users may need to be addressed. If you deploy an accelerated release, you may need to deploy it again when remote computers return, or have them install it before they can connect.

Tools and technologies for deploying a security release are discussed in Part II, Chapter 4, "Security Release." If you do not have a patch distribution infrastructure already in place, consider sending users to Windows Update or Office Update to install a security patch, or putting an emergency Software Update Services Infrastructure in place.

As long as your connection to the Internet has not been disabled during an attack, computers can download and install the software update from Windows Update. This method is also extremely useful for remote users. You can communicate to them the importance of the software update and give instructions for how to use the Windows Update Web site.

As a last resort, should an attack impact network services, consider the manual deployment of a software update. This would involve visiting each computer and applying the release, or providing CDs and installation information to all users in a coordinated manner.

Monitoring and De-escalation

After you have installed a release in the production environment, continue to monitor your computers and network. Watch for a recurrence of any attack signatures determined when learning about the attack. As well as monitoring existing servers, it is important that you monitor the environment as a whole to ensure that new computers added to the network are not vulnerable, enabling the attack to start again.

De-escalation indicates a return to normal business operations. De-escalation typically occurs when none of the parties involved in the incident are identifying or reporting new information.

Post-incident Activities and Review

After the incident is over and the attack is no longer considered an active threat, there are a few wrap-up activities that should be performed, including:

- Submit a change request following your organization's typical change process and re-release any production changes that were required during the attack. If testing was skipped earlier, now is the time to properly ensure that the implemented production changes do not negatively impact the security and reliability of your environment.
- Ensure the vulnerabilities that were exploited are added to you vulnerability scanning reports and security policy computer standards so the attack does not have an opportunity to recur.
- Assess the total incident damage and cost—both downtime costs and recovery costs.
- Review your organization's performance throughout the incident. Take this opportunity to improve your incident response plan.

7

Optimizing Results

Measuring and Improving Performance

After your security patch management process is established and running, you will want to ensure effectiveness, monitor performance, and improve results over time. Even with proper planning, there may be improvements to the process that you can identify through monitoring and assessment.

There are three primary areas of importance within security patch management that you may want to measure and improve upon:

- Improving security releases
- Improving security policy enforcement
- Improving emergency security response

Organizations can monitor, measure, and improve these processes independently by tracking basic statistics and considering the areas described in the following sections.

Alternatively, organizations can optimize security patch management as part of a security assessment or an operational assessment. High-level information on each of these types of assessments is provided at the end of this chapter.

Improving Security Release Response Times

Throughout the process of deploying security releases, there are several points in time that your organization can track to help identify problem areas and improve future performance.

Note: In addition to time-driven improvements, also discuss and assess quality improvements to security patch management. Any time a security patch is improperly assessed, not applied effectively, or applied in a way that causes problems, the problem should be reviewed to determine what needs to be improved.

The following list describes several points in time that should be tracked throughout the security patch management process for each vulnerability:

- T_a = **Awareness** of vulnerability. Early public awareness of a vulnerability.
- T_b = **Bulletin** for a new security-related software update and countermeasures to address a vulnerability.
- T_c = **Change** request approved.
- T_d = **Deployment** of software update confirmed.
- T_e = **Exploit** developed for vulnerability (an actual or potential attack). This is when a virus, worm, Trojan horse, or other known attack tools are released to exploit this particular vulnerability.

If you want, you can track additional data points during change and release management to measure and improve specific activities that are of interest.

Note: Typically T_a=T_b, public awareness of a vulnerability happens at the same time as the software update becomes available. Microsoft® works with the organizations that find vulnerabilities to release software updates at the same time that the vulnerabilities are publicly announced.

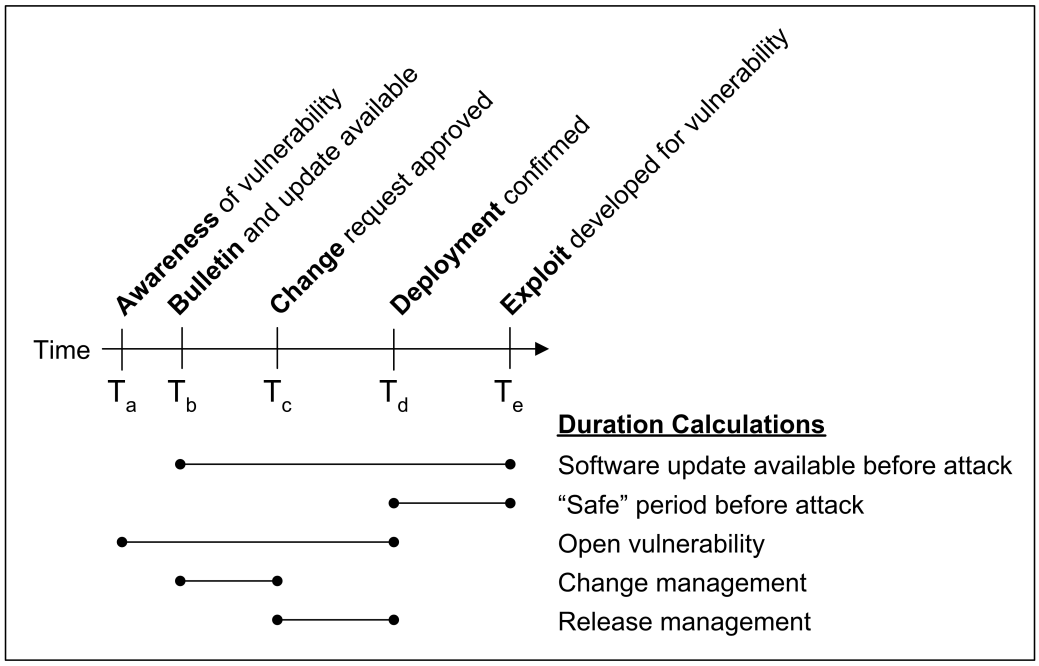


Figure 7.1
Example Timeline with Duration Calculations

The duration calculations presented in the following sections can provide insight into what's working and what isn't during security patch management.

Calculate these durations for each vulnerability, and then determine the averages for the time period being measured and compared.

Look for examples of quick results and slow turnaround times, and use these as a source of both positive feedback and areas for improvement in the future. Also monitor the averages for changes (ideally, improvements) over time.

In addition to time-driven improvements, also discuss and assess quality improvements to security patch management. Any time a security patch is improperly assessed, not applied effectively, or applied in a way that causes problems, the problem should be reviewed to determine what needs to be improved.

Software Update Availability Before Attack, T_e-T_b

This calculation describes the typical situation of a bulletin, software update, and related countermeasures being available (T_b) before an exploit (T_e) is developed and an attack can take place.

It is important to track how quickly vulnerabilities are exploited by viruses and other forms of attacks over time. This duration is out of your organization's control, but it can be used to influence the recommended timeframes that are in your control.

Although past timeframes are no guarantee of future attack timeframes, understanding this information is a good source of input when setting realistic and useful release targets for future vulnerabilities.

If this duration decreases over time, you may want to consider reducing your recommended software update release timeframes that were introduced in Table 4.1: Security Release Deployment Timeframe Guidelines in Part II, Chapter 4, Security Release.".

Note: For examples of this duration for several historical attacks, see Table 1.7 Historical Attack Examples and Related MSRC Bulletins in Part I, Chapter 1, "Introduction to Security Patch Management."

"Safe" Period before Attack, T_e-T_d

This calculation describes the period of time after the software update has been deployed (T_d) and before the exploit (T_e) is developed and an attack can take place. You should try to increase the average safe period over time.

If this duration is always positive, it means that your security patch management process has been effective at releasing security patches before the vulnerabilities have been exploited. Use these as examples of how security patch management has proactively avoided problems before they occurred.

If this duration is ever negative, you have a problem that needs to be addressed—your organization is either reactive and waiting for problems to occur, or too slow when performing security patch management.

Note: Vulnerabilities can recur for many reasons after software updates are distributed, meaning that it is possible for a "safe" environment to become vulnerable. See the following section on Improving Security Policy Enforcement for more information on improving performance with the recurrence of historical vulnerabilities.

Open Vulnerability Duration: T_d-T_a

This calculation describes the period of time it takes your organization to deploy a software update (T_d) after learning of a vulnerability (T_a). You should try to decrease the average open vulnerability period over time.

This is the period of time during which your organization is vulnerable to an attack for this specific vulnerability. Compare this calculation with the typical software update availability duration before an attack. The goal should be to reduce open vulnerabilities more quickly than they are actively exploited.

Use the following calculations to more accurately understand where time is spent during security patch management and improve as required.

Release Management Duration: T_d-T_c

This calculation highlights how long an organization spends on various activities during release management.

To improve release management, consider each of the following areas:

- **Release planning.** Should the change request form include additional information that would help streamline release planning?
- **Release development.** Are there tools or skills that your organization needs to have to improve their performance during this activity?
- **Acceptance testing.** Are there tools that could improve testing timeframes? Is the test environment properly staffed and managed?
- **Rollout planning and preparation.** Does your organization have enough information about the environment to deploy patches effectively? Is there additional computer information that could be tracked regularly?
- **Release deployment.** Are you using tools that help with release deployment, or is it primarily a manual process?
- **Change review.** Are there sufficient reports and information collected that can report back on the success or failure of a release?

Change Initiation and Management Duration: T_c-T_b

This calculation highlights how long your organization spends on the change initiation and management process.

To improve change management, consider each of the following areas:

- **Change initiation.** How long does it take to determine that a vulnerability and associated security patch is appropriate for your environment? Does your organization have the inventory information that is required to quickly and accurately determine relevance?
- **Change classification.** Is it easy to determine the priority of a security release? Is the required inventory and configuration information available, or easy to collect? Do you have tools that provide this information?
- **Change approval.** How quickly are changes approved? Does the change approval team meet frequently enough? Do they have all of the information necessary to make a decision?

Improving Security Policy Enforcement

Even after vulnerabilities have been addressed within an environment, they can recur. New computer installations, reinstallations, and distributed or end-user administration can contribute to the possibility of a vulnerability recurring and exposing your entire infrastructure to attack.

Monitor the vulnerability scanning reports over time, and create an incident each time a vulnerability recurs in the environment (this is best done through an incident management tool). Regularly monitor the number of outstanding security incidents and the total number of security incidents.

When vulnerabilities recur regularly in an environment, this is usually a symptom of a larger administrative issue or a security policy issue. To address this issue, consider some of the following areas:

- **Baselined computer images.** Does your organization have baselined computer images to standardize and accelerate the installation process? Does everyone use these images when installing new computers in the environment? Are the images (or image post-installation instructions) updated with new security patches?
- **Decentralized administration.** How many groups in your organization install and maintain computers? Do they each follow the same security policy or guidelines? Do they adhere to the same change and release management standards? Is there accountability for safe computing practices within each group?
- **Administrative access.** How many people in your organization have administrative rights to install software? Are all end-users administrators? Is each administrator aware of your evolving security policy? Should this policy be reconsidered?
- **Common sources of security issues.** Do most of the security policy violations come from a particular department, or on a particular type of computer (a certain model, operating system, or application)? Perhaps additional training or special attention in a particular area is required.

Changing the policies, procedures and automation that apply to the administration of computers can significantly reduce the number of recurring vulnerabilities in your environment.

Aside from security policy and administrative changes, you can improve incident response times to security issues through improved escalations and other areas such as forced patching. These are discussed in Part II, Chapter 5, "Enforcing Security Policy."

Improving Emergency Security Response

Ideally, organizations never have to implement their contingency plans and procedures for an emergency security response in a real attack situation.

Should the unfortunate situation arise in which an attack response is necessary, keep in mind that this provides an ideal opportunity to learn and to improve from the experience. After the event, spend time with key people from each impacted group to reflect on the attack, how your organization progressed through the experience, and what could be done to improve future results.

Alternatively, your emergency security response plans can be reviewed internally or by a third party, or you might consider doing a trial run of emergency security response procedures by staging or simulating an attack. You might even consider using penetration testing to find weaknesses and determine the effectiveness of your intrusion detection reporting and resulting emergency security response procedures.

Emergency Security Response Review

When reviewing your organization's performance at handling an attack (whether staged or real), try to create a no-blame environment in which the focus can be on improving future performance by improving process and procedures. The agenda should minimally cover the following elements:

- **Incident timeline.** What happened when? It is difficult to discuss an event without having an accurate view of what happened from a few different perspectives.
- **Success factors.** What went right? What aspects of the process are working and should be encouraged and repeated?
- **Areas to improve.** What went wrong? What happened that didn't work the way it was intended? What was missed completely?
- **Suggestions for change.** What should be changed? How will you be better prepared for a future attack, and who is responsible for making this change?

Ideally the review will take between four hours and one to two days. Do not schedule the review immediately after an attack—instead, wait a couple of days for events and emotions to calm down.

Send out pre-work materials to make the review meeting more effective—either a short survey to be completed in advance or a list of areas that people should come prepared to discuss.

When reviewing an emergency security response consider the following elements:

- What should be done in the future to prevent attacks?
- How is an attack identified? How is the rest of the organization informed of the situation? How does each group get the information it needs?
- What additional contingency planning should be prepared for a future attack?
- Does each group understand its roles and responsibilities during an attack? How are different groups expected to work together?
- How can you communicate effectively and comprehensively, considering that some communication channels may be impacted?
- How are end-users appropriately informed during the event? How are they trained and prepared before an event?
- Could someone take advantage of the confusion that occurs during an attack to gain additional access? How could this be prevented?

Be sure to prioritize the results that come out of the review and have clear owners and timelines next to each action item. Schedule a meeting at some point in the future to review and share progress toward accomplishing these goals.

Operations Assessment

Security patch management is one of many areas of IT operations. Organizations spend a considerable percentage of their IT budgets on operations, because achieving operational excellence improves expenditures while improving mission-critical service reliability, availability, supportability, and manageability.

An operations assessment enables an operations staff to realize tangible benefits to existing or proposed operations, regardless of the size of the enterprise or its maturity level.

For a quick self-assessment of your organization's operational excellence, see the Microsoft Operations Framework Self-Assessment Tool: http://www.microsoft.com/technet/itsolutions/tandp/opex/moftool.asp.

To learn more about operational assessments and to find consulting services that can perform an operations assessment, see the Operations Assessment Service Offering: http://www.microsoft.com/solutions/msm/evaluation/overview/opsassessment.asp.

Microsoft Services performs these assessments on a regular basis.

Security Assessment

A security assessment is intended to help security professionals develop a strategy to protect the availability, integrity, and confidentiality of data in an organization's IT infrastructure.

A typical security assessment includes organizational assessment, asset valuation, threat identification, vulnerability assessment, and security risk assessment. The results include a security action plan and security risk contingency plan that should be implemented in the environment, including operational modifications and improvements.

A security assessment may be conducted at the organizational level, or on a subset of the environment.

For more information about helping an organization perform its own security assessment, see Security Risk Management Discipline: http://www.microsoft.com/technet/security/prodtech/windows/secwin2k/03secrsk.asp

The Microsoft Gold Certified Partner Program for Security Solutions identifies several companies that have experience with security technology and performing security assessments. For more information on this program, see the Microsoft Gold Certified Partner Program for Security Solutions: http://members.microsoft.com/partner/partnering/programs/securitysolutions/default.aspx.

Index

P

Q

R

S

T

U

V

W