realtimepublishers.com™

The Administrator Shortcut Guide™ To

Patch Management

NEW BOUNDARY
TECHNOLOGIES

Rod Trent

Copyright Statement

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

Chapter 2: Patch Management Best Practices

Several companies and security patch administrators consider the patching process to be a single step that provides a secure computing landscape. In reality, the patching process is a continuous cycle that must be strictly followed. Each step in the process must be tuned and modified based on previous successes and failures. As many realize, patching computers is a fact of life as part of the defense in depth security strategy. By spending time up front to create policies and procedures, companies can minimize the time and resource requirements needed to fulfill the patching demands.

In this chapter, you will read about each step in the patch management process. Throughout this discussion, keep in mind that each step can only be performed successfully in the future if the lines of communication are clear and each step is documented accurately. Documentation and communication are critical to the patch management process.

In many companies, those entrusted with the task of securing the environment by distributing patches have many other jobs heaped on them—they perform double-duty by not only being patch administrators but also managing the company's email system or network and performing myriad other tasks. The number of patches released each month makes patching into a full-time job; thus, those that are responsible for tasks in addition to patching can feel overwhelmed. By employing the right technologies and developing and implementing the proper patch management processes for the environment, patch distribution can be seamless. In addition, keeping a strict regimen to the patching process can make patching an almost automatic task.

The processes outlined in this chapter are merely guidelines; they are based on collective industry standards for patch management. In developing your patch management process, you need to review your environment and use this assessment to develop appropriate strategies. Each computing environment is different, but the processes in this chapter give you a framework for building your own guidelines to make your computing environment secure.

Because patch management is designed to give an organization control over the software updates it deploys, any organization planning to patch its operational environment should ensure that the company has:

- Effective operations, including people who understand their roles and responsibilities
- Tools and technologies that are appropriate for effective patch management
- Effective project management processes

There are a few terms that you need to be aware of as you read through this chapter. Table 2.1 lists the key security terms used in relation to the patch management process.

Term	Definition
Vulnerability	Software, hardware, a procedural weakness, a feature, or a configuration that could be a weak point exploited during an attack; sometimes referred to as 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

Table 2.1: Patch management–related security terminology.

Prerequisites for the Patch Management Process

Many guides on patch management jump straight into the patching processes, leaving you with very little understanding of how to incorporate the processes into your own environment. Such guides don't give the reader a starting point. Rather than jumping in without establishing this basic knowledge, let's explore prerequisites that should be observed and how the processes will ultimately apply to your company's needs.

There are things you need to know about your environment before you start throwing policies and procedures at management for approval. There are several levels of tasks that you need to have a handle on before distributing patches to the end users and expecting them to adhere to an iron-handed approach for computing security. Skipping these important aspects can make your patch management processes unsuccessful from the start as well as cause management to question whether patching computers is a worthwhile investment of time, resources, and technologies. The prerequisites covered in this section are:

- Know your computing environment
- Prepare end-user education
- Assign responsibilities
- Understand the current process
- Develop a chain of communication
- Baseline
- Acquire management buy-off

Know Your Computing Environment

Knowing your computing environment may sound like a simple task, but it entails more than simply having an inventory. You might already be aware of the hardware and software in use within the company, but there are a few more factors to consider than just the installed computing devices. In addition to the knowledge of the computing equipment and software that makes up your computing landscape, you need to have full knowledge of the following factors. Ask yourself the following questions to determine your level of understanding of your own environment:

- IT staff security knowledge
  - How security adept are the other members of your IT support team? Are they as knowledgeable about good computing security as you are?
  - Do they practice the same security procedures with their own equipment that they would with an end user's computer?
  - Has anyone on your team taken classes or training to better acclimate themselves with the current security landscape?
  - If an attack on your network happened overnight, would you feel comfortable that other members of your team would know the proper steps to mitigate the attack?
- Adequate resources
  - Are you the only one tasked with patching the computers in your organization or are there others on your team that can help?
  - Is there tension between you and others on your staff or can you work side-by-side to deploy patches quickly before the next big attack?
- End user knowledge and comfort level
  - Are the end users you service comfortable with their computing environment?
  - Have you educated the end user population about the risks involved with leaving a computer unpatched?
  - If an attack occurred, would the end users panic?
  - Does the end user population understand the importance of patching their computers, keeping up-to-date with their antivirus software, being wary of strange emails and attachments, and using personal firewalls?
- Building the infrastructure
  - Can you deploy patches quickly in your current network infrastructure?
  - Do you have users who dial-in to the network regularly and who rarely visit the office?
  - Are your server resources adequate for employing a patch management application?

Prepare End User Education

Another prerequisite for implementing a patch management process is to determine the level of expertise within your end user population and create some type of company standard communication. Give end users the information they need to understand your patch management policies and how patch management can affect the company's profitability and their own productivity. Unless the end users are completely aware of your policies, you'll have a tough time deploying patches successfully. Depending on the technologies you deploy for distributing patches, the end users might decide it's not worth installing a patch if it means a few minutes of lost productivity—or they might choose to terminate the installation or delay it to a later time.

> ✎ It is a good practice to make your patch management communication different than normal enterprise software distribution notifications because patching is more serious and should not be disregarded. Prepare your communications so that they grab the attention of the user population.
>
> Given the damage that one unpatched machine can do to an organization, allowing end users control over accepting a patch can be a high-risk decision. A patch tool should allow you to automate a silent install of the patch.

When developing end user training, you can help yourself immensely by generating specific training for upper-level management. You should help them understand that patching the environment protects the company from data loss, lost productivity, and a loss of revenue.

Assign Responsibilities

Based on the patch management phases described later in this chapter, assign responsibilities for the tasks you require to implement the patch management policies. Although you can automate many tasks by using a good patch management application, there are many tasks that you will still need to manually perform. Assigning these responsibilities up-front will give your team members a sense of ownership and perspective on how they fit into the overall security of the computing environment.

Understand the Current Process

What does your current patch management strategy look like? Are you walking from computer to computer to install patches manually? Have you already employed a self-constructed mechanism to deploy patches? When you start to employ the patch management process, in most cases, you'll want to retrofit the process to your current procedures. In other cases, you might need to drop your current processes completely and start from scratch. Understanding your current process will allow you to develop a plan of action for incorporating your new knowledge.

Develop a Chain of Communication

When a patch has been tested and is ready to be deployed into your production environment, you can increase the likelihood of success by providing clear communications to all those involved with the deployment. Developing a chain of communication before implementing your patch management process will help your overall planning and policy development. There are three categories of groups that you need to take into account when communicating the pending deployment of a patch:

- **The patch management team**—When determining this team's membership, consider adding representatives from the following groups (if they exist in your organization—in some cases, a single individual might provide many of these services):
    - Help desk team
    - Network team
    - Security team
    - Application deployment team
    - Desktop deployment and imaging team
- **Management**—Communicating to the company's business management is important so that they can help alleviate problems with end user complaints. Getting management on your side is a huge boost for your patch deployments. Because most business managers are not technically savvy, make them understand the process in a way that gives them the assurance that you are doing the right thing and that employee productivity and company stability is your goal.
- **End users**—The end user is the ultimate target for the deployment of patches and can make or break a successful deployment. Be sure to properly word your end user communications based on their computing comfort level so that they are fully aware of the pending patch installation. Most end users feel that the computer they are using is their own (instead of a company-purchased and -owned device) and they treat it as such. If you go the extra mile to ensure that end users understand that you are helping them instead of attempting to hinder their productivity, they will be more likely to help you in making the patch deployment successful.

Baseline

A baseline is a set of configurations for a product or system that has been established as the company standard for building and deploying systems. An application or software baseline should contain the information required to rebuild a system to a desired state. And, more importantly, the baseline should be used to rebuild or deploy a new system to the most current secure state—meaning that the baseline should contain all of the most current vendor-released patches. Your environment might require several separate baselines to meet the needs of the organization. For example, the HR department will need a different set of applications installed than the engineering group will require. Thus, each department would need a separate baseline.

Baselining is also part of the assessment phase of the patch management process described later in this chapter. Although this process is part of the assessment phase, baselining is also an important prerequisite before applying the processes. If you can develop a secure baseline before applying your patch management processes, you can ensure that your environment is secure prior to deploying new patches. Doing so will also help mitigate older vulnerabilities as well as decrease the need for future patching. As new patches are released and deployed throughout your organization, and new computers are deployed or old ones are rebuilt, the baseline must be updated to reflect the latest level of security.

📖 The process of updating the baseline can be handled through the standard patch management processes, which we explore later in the chapter.

Adhering to your baselines is important because a single unpatched, non-standard computer can make the entire environment vulnerable to attack. If a computer is deployed that falls under your current baseline, you will have to act quickly to bring it into compliance—providing you have the tools to know when a suspect computer is connected to your network. If you don’t have the tools in place to monitor suspect computers, you may never know until a worm (that you thought you patched against several months ago) starts spreading through your network.

There are several approaches to the process of building and maintaining baseline configurations. One approach is to use a disk imaging (also referred to as *cloning* or *ghosting*) application. A disk imaging application performs a sector-by-sector copy of the contents of a hard disk to an *image file* that can then be copied to the same or a different hard disk. The disk image contains the contents of the hard disk—the OS, drivers for the particular hardware configuration, applications, patches, and so on.

If your organization has standardized on a single OS and maintains a fairly homogenous hardware environment and most or all users run the same applications, a disk imaging tool alone might be sufficient for your baseline process. However, in more complex environments in which there are multiple OSs, multiple flavors of PCs (which each require different drivers to be installed), and/or different applications being used by different groups or individuals in the organization, a disk imaging tool becomes impractical for your baseline process. The number of images you need to maintain grows exponentially, and each week you would need to rebuild dozens or possibly even hundreds of images to add the latest security patches.

In this event, disk imaging coupled with your software deployment tool might prove to be a better overall option. In complex environments, maintaining baseline configurations can be accomplished by combining a disk imaging tool with your software deployment tool. A disk image for each base OS and hardware configuration combination is maintained (the image contains the base OS, the drivers for the target PC hardware configuration, and the core applications run by all users). When a PC needs to be built or rebuilt, the appropriate image is installed to lay down the base OS and core applications, and then your software deployment tool is used to install the applications required by the user of the target PC and all of the necessary security patches.

Management Buy-Off

One of the most important aspects of the patch management policy you develop is support. Support can come from many places, but the key area of support is from the business management group. Management must be included in all aspects of your patch management planning and policy building. Ultimately, you need to create policies that everyone agrees upon and that gain management approval. Once management approves of the policies you have submitted, you can rely on management for support and to be the iron hand that helps enforce the policies. If the end users know that management is behind you in your quest to secure the environment, the end users will be less likely to challenge the methods you utilize to accomplish the task.

The Patch Management Process

The patch management process can be broken down into four distinct processes, with specific tasks assigned to each phase. The four phases are:

- Assess
- Identify
- Evaluate and Plan
- Deploy

Although these bullet points appear to be steps in a linear process, they are actually recurring events. The patch management process needs to be ongoing even when new patches are not available. As you'll see when reading through the phase descriptions, there is much that is done to prepare for future patch releases.

Particularly for those tasked with many jobs, incorporating the patch management process into an already heavy daily workload sounds like a daunting task; however, such does not have to be the case. With proper planning, the necessary tasks can easily be undertaken in such a way that they become second nature and are easily accomplished. Critical to successfully implementing a patch process in any size organization is the selection of the right patch tool. Patching tools should include features that help automate many of the tasks in the patch process.

> 📖 Chapter 3 will describe the capabilities and features of available patch management tools, allowing you to determine which set of capabilities best fit your environment, particularly when budget is a concern.

Figure 2.1 provides an illustration of the patch management process phases. This figure gives a better understanding of how the four phases of the patch management process flow. As I mentioned earlier, patch management is a circular process and must be ongoing.

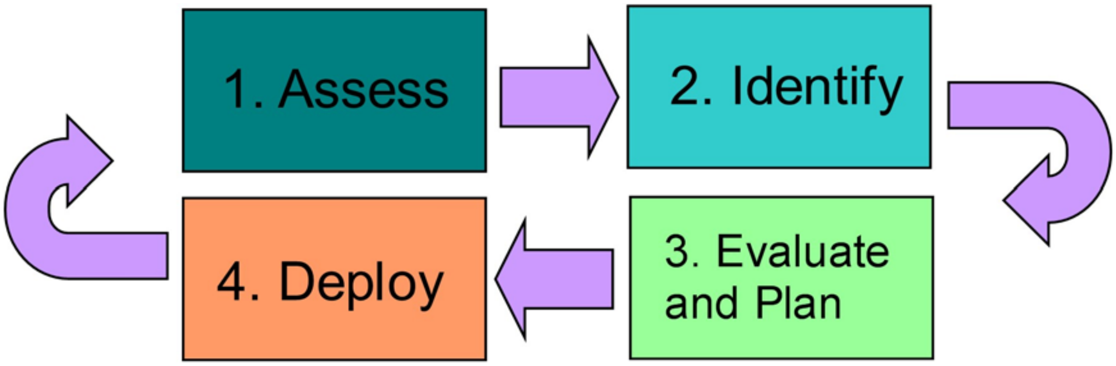

Figure 2.1: The four patch management phase workflow.

Assessment Phase

The patch management process starts with an assessment of what you have in your production environment, what security threats and vulnerabilities you might face, and whether your organization is prepared to respond to new software updates. Long before a new patch is released, you should be preparing your environment for potential deployments. Gathering information about your environment, or *assessing* your environment, gives you the knowledge you will need to deploy patches successfully.

Inventory

Only by employing some type of inventory mechanism can you be sure that your environment meets the recommended specifications for security. The application you choose to manage your inventory should be able to gather information about your environment that provides answers the following questions:

- How many OSs are present that will potentially need to be patched?
- How many versions of the OS are in use?
- How many applications and application versions are in use?
- Will different OS or application versions need different patches?
- How many unpatched systems are being used, and which ones are they?
- How many unmanaged systems are being used, and which ones are they?
- How many and which systems are mission critical?
- What existing software dependencies will impact how a patch is distributed?

Baselining

Mentioned previously in this chapter as a prerequisite to patch management, baselining requires that you perform updates to maintain standard deployments for the computers and services in your environment. Thus, when you research technologies to help with your patch management processes, look for products that provide baseline-incompatibility alert functionality (products that let you know if a system no longer is configured to the appropriate baseline configuration). Keep in mind the following points when establishing baselines for your environment:

- You should bring all the computers that have been identified as below the baseline up to compliance. These computers may have had issues with distribution, schedules, or permissions, or may require special care through exception handling.
- Computers exceeding their class baseline should be checked to determine whether unauthorized changes have occurred. (Systems that exceed an approved baseline contain application versions or software updates that have not been tested for interoperability and formally approved by IT operations and security.) In some cases, it might be necessary to return a system to a trusted level or control it through a change freeze.
- Some systems may have special circumstances that make them exempt from the baseline. For example, an older workstation running a legacy payroll application that connects to a processing agency by means of a modem may require an OS level far below the established baseline. It may not be appropriate to upgrade this system to the latest baseline because doing so could prevent the legacy application from running. You can mitigate the impact that exceptions like this will have on your network by placing these systems behind a firewall or on a separate VLAN and ensure that they are used to perform only the business tasks for which they are necessary (those tasks that keep them below the baseline).
- You can use scanning technology such as the Microsoft Baseline Security Analyzer (MBSA), New Boundary's Prism Patch Manager, or Shavlik's HFNetCheck to help establish baselines and determine whether the computers in your environment adhere to set baselines. Many times, it's a good practice to use multiple toolsets as part of baselining in order to avoid false positives.

📖 For more information about MBSA, see http://www.microsoft.com/technet/security/tools/mbsahome.mspx.

Like the entire patch management process, the assessment phase is ongoing. You need to always know what computing assets you have, how you can protect them, and how you can ensure that your software distribution architecture is able to support patch management. The tasks for providing ongoing assessment are:

- Inventory existing computing assets
- Assess security threats and vulnerabilities
- Determine the best source for information about new software updates
- Assess the existing software distribution infrastructure
- Assess operational effectiveness

Identification Phase

Your goal during the identification phase is to discover new software updates in a reliable way, determine whether those updates are relevant to your production environment, and determine whether an update requires a normal-process or emergency deployment.

The first step in the identification phase is to know when patches are available. How are you currently notified when new patches are released? If you do not already receive the email alerts from Microsoft pertaining to patch releases, you should sign up at http://www.microsoft.com/technet/security/bulletin/notify.mspx for the email security alert.

There are also plenty of other resources that provide email services and other methods of notification when new patches are released. These other resources (listed at the end of Chapter 1) also provide notification alerts for platforms and applications other than those from Microsoft. In some cases, the third-party alerts might provide more information than the Microsoft alerts. Identifying when new patches are released is crucial to knowing when to rev up the patching engines.

> Regardless of how familiar you are with patch-notification emails, always verify the source of the messages. When new patches are released, virus writers send infected emails in droves to try to catch unsuspecting and unknowledgeable recipients. Although an email may appear valid, such is not always the case. Be very wary of any email you receive that announces the release of new patches, particularly those that include the patch as an attachment to the email. Microsoft NEVER sends patches through email. If you receive an email, seemingly from Microsoft, with an attached “patch,” delete the message immediately. You should include this information in your end users’ education.

Once you’ve received notification about new patches, review the Microsoft Security Bulletin associated with the patch. In Microsoft’s Security Bulletins, important information about the patch is made available and is broken down into logical sections as shown in Table 2.2.

Section	Description
Summary	The Summary section is a quick read meant to help you immediately understand the Maximum Severity Rating, Impact of Vulnerability, Affected Software, and Recommendation items. This information can assist you in determining how relevant the patch is to your environment. The Summary should be the first section you read.
Technical Details	The Technical Details section provides an in-depth technical description of vulnerabilities. This section also outlines the mitigating factors and the severity of vulnerabilities for all affected products.
Workarounds	The Workarounds section gives you potential workarounds to mitigate any threat until you are able to patch your environment.
Frequently Asked Questions	The Frequently Asked Questions section provides answers to commonly asked questions specific to the vulnerability or fix that the patch has been developed to address.
Security Patch Information	The Security Patch Information section lists items such as prerequisites, platform-specific installation information, deployment information, restart information, removal information, file information, and patch verification steps.
Knowledge Base article	The Knowledge Base article section refers you to the Microsoft article associated with the security bulletin. The article provides additional information about the vulnerability. The number in parentheses to the right of a security bulletin's title can be searched and found at http://www.support.microsoft.com/.

Table 2.2: Microsoft Security Bulletin sections.

Spend time reading the security bulletin and associated Microsoft article. Doing so is critical to helping you understand whether the patch applies to your environment and how you should classify the patch for deployment.

> 🕮 If you are interested in learning about the security bulletin release process that Microsoft has developed, read the “Revamping the Security Bulletin Release Process” article at http://www.microsoft.com/technet/security/bulletin/revsbwp.mspx.

Once you understand the details about the patch and you have identified that the patch applies to your organization, download the actual source files. Most patch management applications will download patch source files for you either based on a set schedule or through an updating mechanism or wizard. Either way, get your hands on the files needed to start testing the patch.

Prior to full-blown testing, verify that the files are good and that they install and uninstall correctly, as prescribed in the security bulletin. Review all the options available for deploying the patch, and document these options before entering the evaluation and planning phase. If different individuals perform the identification phase and evaluation and planning phase, the documentation created during the identification phase will be extremely useful to those performing the evaluation and planning phase.

Evaluation and Planning Phase

The goals during the evaluation and planning phase include:

- Make a go/no-go decision to deploy the software update
- Determine what is needed to deploy the update
- Test the software update in a production-like environment to confirm that the update does not compromise business-critical systems and applications

A big part of evaluation is to determine the importance of releasing a patch. Some patches may apply to only a small area of your computing environment; others might affect your entire organization and require that you deploy with the utmost speed. Develop a patch classification system that will allow everyone to understand the critical or non-critical needs for each patch. Each organization's classifications and associated deployment time frames will vary based on the organization's needs; Table 2.3 provides a framework for building your own classifications and time frames.

Classification	Recommended Deployment Time Frame
Critical	Start deployment within 24 hours
High	Start deployment within 1 week
Medium	Start deployment within 1 month, or opt for a service pack or update rollup
Low	Opt for a service pack or update rollup

Table 2.3: Patch classification and associated deployment time frames.

The "Start deployment within…" wording actually indicates when the patch should be deployed into your production environment *after* testing. As shown, your testing procedures will need to be optimized to ensure quick deployment should the patch be classified as critical. You will need to develop different testing procedures depending on the level of classification you give the patch. For example, a classification of critical will require a minimal, but rigorous testing cycle; a classification of high will allow for a potentially longer testing cycle than patches deemed critical; the medium and low classification testing process would be a longer, more "at your leisure" testing cycle.

Once you have successfully classified the current patch release, you can begin testing in an environment that closely resembles your production environment. Many companies cannot afford the luxury of a complete testing lab that is used purely for patch testing. A lot of these companies have employed virtual machines to provide the required testing environment for which to test the patches. Although this setup can, in most cases, give you an adequate test bed for verifying that a patch will install and work with your company's installed set of applications and services, you should be wary of this setup. A recent patch tested in a virtual environment worked well, but when deployed in production, caused Windows NT 4.0 computers to blue screen. If you use virtual machines to perform testing, make sure to also test on a few production systems to verify the accuracy of your virtual machine results. Thus, you would need to test the package in your lab environment (real or virtual), then pilot test it in a production environment to confirm that it does not compromise your business applications.

During your testing, use the information passed on from the identification phase to fully test all of the patch's options for installation and uninstallation. Create a plan for how the patch should be deployed (for example, workstation must be logged on, patch must be deployed silently, a reboot is required for the patch to install successfully) as well as how the patch can be uninstalled (aka *rolled back*) should the deployment hit a snag. If different individuals will be performing the deployment than those who are testing the deployment, the documentation and deployment plan created by the testers will be useful to those assigned to deployment.

Deploy Phase

The goal during the deploy phase is to successfully roll out the approved software update into the production environment in such a way that you meet all of the requirements of any deployment service level agreements (SLAs). Once you have followed through all the previous phases, and you know the patch is ready to deploy across your production environment, there are three activities that you must perform to complete the deploy phase:

- Deployment preparation
- Deployment of the patch to targeted computers
- Post-implementation review

Deployment Preparation

For the deployment preparation, notify everyone in your defined communication channel that the patch is pending deployment. As described at the beginning of this chapter, this communication channel should be a group of representatives from the various areas of your company. You will want to communicate that the patch is ready for deployment and relay the patch deployment schedule. Give the representatives in your communication chain enough time to pass on your communication to their respective areas.

In the communication, make sure to give enough information about the patch deployment to make the rollout a success. Give clear instructions on when the patch will arrive at the computers (have end users leave their computers on if the patch will arrive overnight) and enough instruction to allow each computer user the necessary tools to install the patch correctly (if it has not been distributed to run silently). Also, make sure to indicate the importance of the patch. Some end users may want to delay distribution due to critical business functions. If you include the case for the deployment in your communication, end users will be less likely to question your judgment.

Once the communication has been relayed and you've given it enough time to proliferate across the necessary groups, stage the patch so that it is ready to deliver. Depending on the technology you utilize for deploying the patches to the workstations, you should prepare the delivery mechanism and configure the schedule.

> ✎ Communication is the most important aspect of deployment preparation. By giving enough time for the communication to be relayed through your communication chain and on to the various groups, you allow those with good reason to gain approval for delaying the deployment of the patch due to business requirements or to work with you to revise the schedule to a more convenient time.

Deployment of the Patch

Obviously, deploying the patch using your technology of choice is the next logical step in the process. But there is also much more to this step than simply “throwing the switch,” crossing your fingers, and hoping the patch finds its way across your environment. In addition to initiating the deployment, you will want to:

- Monitor and report on the progress of deployment
- Handle failed deployments

If you employ home-grown patching solutions or vendor offerings that don’t provide real-time monitoring and reporting of patch deployment progress, you will benefit greatly from investing in a product that offers this functionality. The ability to monitor and report on the progress of the patch deployment (which, incidentally, includes information about whether the patch made it to the computer, whether the computer attempted to install the patch, and whether the installation was successful) enables you to identify those computers for which assistance may be needed. By utilizing technologies that allow you to monitor and report on the full patch deployment progress, you can seriously minimize the amount of time you need to troubleshoot failed installations. Otherwise, you will have to wait for the end user to contact you about a failed installation. Some end users never bother, which leaves your environment in an undesired, unsecured state.

> 📖 For information about how Microsoft patches its own computers, read “Windows Patch Management: How Microsoft Patches its Own Client PCs” at http://www.directionsonmicrosoft.com/sample/DOMIS/update/2004/01jan/0104cpam.htm.

If a deployment is unsuccessful and must be rolled back, you must have a plan in place as part of your patch management process to stop the rollout, uninstall failed updates, and redeploy them. This plan should have been identified during testing in the evaluation and planning phase.

Post-Implementation Review

The last step in the deploy phase is to gather your deployment statistics, discuss them with your patch deployment team, and document them. Use these statistics to determine:

- Whether the deployment was successful
- Whether you need to tweak any of your processes to ensure better success in the future
- The performance of those individuals with specific tasks in the process
- Whether any SLAs need to be adjusted

And, most importantly, create or update your baseline to prepare for the next assessment phase.

Patching Challenges

There are patching challenges in any company, and each company will have its own set of problems and hurdles to surpass. Most of these challenges can be alleviated by deploying the right technology for your environment, or modifying your patching policies to meet your company's needs. There are a couple of specific challenges that you need to keep in mind when building your patch management process: remote users and foreign connected computers.

Remote Users

Remote workers are a challenge for any company. As time progresses, more and more employees work from the road or from home. This situation presents a big challenge for companies that are trying to bring the entire computing environment within a secure operating level. If you have a large number of remote users that you must manage, when researching technologies to automate patching, make sure to earmark those technologies that offer some sort of remote computer management. There are a number of products on the market that make better use of slow connections for deploying patches and software.

> 📖 Chapter 3 will explore these technologies in more detail.

Foreign Connected Computers

The term "foreign connected" actually indicates those computers that connect to your network that are not *owned* by your company. These computers are generally computers that require a connection to your network to do specific tasks that have been authorized by management. Some examples of these computers are:

- Contractors
- Customers
- Vendors
- End users' home computers

These computers still need to be managed by you, or at least, the computers need to be reviewed by you before gaining authorization to connect to your environment. You might not need to manage the patching levels of these computers, but you need to make sure that these systems will not become a threat agent to your environment. You'll want to include information about this scenario in your patch management policy.

Microsoft Terminology for Software Updates

The Table 2.4 lists the current Microsoft standard terms for software updates. This terminology became effective June 30, 2003.

Term	Definition
Security patch	A broadly released fix for a specific product that addresses a security vulnerability. A security patch is often described as having a severity, which actually refers to the Microsoft Security Response Center (MSRC) severity rating of the vulnerability that the security patch addresses.
Critical update	A broadly released fix for a specific problem that addresses a critical, non-security–related bug.
Update	A broadly released fix for a specific problem that addresses 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 Quick Fix Engineering (QFE) 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 that are released as a cumulative offering and are targeted at a single product component, such as Microsoft Internet Information Services (IIS) or Microsoft Internet Explorer (IE). An update rollup allows for easier deployment of multiple software updates.
Service pack	A cumulative set of hotfixes, security patches, critical updates, and updates that have been released since the release of the product. Service packs include solutions to 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 are tested by Microsoft more than any other software updates.
Integrated service pack	The combination of a product and a service pack in one package.
Feature pack	A new feature release for a product that adds functionality to the product. Feature packs are usually rolled into the product at the next release.

Table 2.4: Microsoft software update terminology.

Patch Management and Security Resources

The following list represents some of the best resources for learning about patch management security.

Web Sites

Microsoft Security Bulletin Search at http://www.microsoft.com/technet/security/current.aspx

Archive of Microsoft Bulletin Release Summaries at http://www.microsoft.com/technet/security/bulletin/summary.mspx

Microsoft Security Bulletin Notification Service at http://www.microsoft.com/technet/security/bulletin/notify.mspx

Microsoft Security and Privacy Policies at http://www.microsoft.com/technet/Security/topics/policy/default.mspx

Information on Bogus Microsoft Security Bulletin at http://www.microsoft.com/technet/security/news/bogus.mspx

Register for the Microsoft Security Newsletter at http://www.microsoft.com/technet/security/secnews/default.mspx

Archived Microsoft Security Chat Transcripts at http://www.microsoft.com/technet/community/chats/trans/default.mspx#XSLTsection139121120120

Archived Microsoft Security Web Casts at http://www.microsoft.com/technet/community/webcasts/default.mspx#XSLTfullModule123121120120

Microsoft Learning Security Resources at http://www.microsoft.com/learning/centers/security.asp

Microsoft Security Checklists and Resource Guides at http://www.microsoft.com/technet/security/chklist/default.mspx

Enterprise Security Tools at http://www.microsoft.com/security/guidance/tools/default.mspx

RSS: Really Simple Syndication Microsoft Bulletin Security Feeds at http://www.microsoft.com/technet/security/bulletin/secrssinfo.mspx

Report a Security Vulnerability to Microsoft at http://www.microsoft.com/technet/security/bulletin/alertus.aspx

PatchManagement.org hosted by Shavlik at http://www.patchmanagement.org/

United States Department of Energy Cyber Solutions Tools Center at http://www.ciac.org/cstc/

Network security articles and hacking prevention resources for the government and general public at http://www.governmentsecurity.org

SecurityFocus.com (hosted by Symantec) at http://www.securityfocus.com/

Articles

“Revamping the Security Bulletin Process” at http://www.microsoft.com/technet/security/bulletin/revsbwp.mspx

“Understanding Patch and Update Management: Microsoft's Software Update Strategy” at http://www.microsoft.com/security/whitepapers/patch_management.asp

“Standardizing the Patch Experience” at http://www.microsoft.com/technet/security/topics/patch/stdpatex.mspx

The NIST Patch Management white paper at http://csrc.nist.gov/publications/nistpubs/index.html and http://csrc.nist.gov/publications/nistpubs/800-40/sp800-40.pdf

Microsoft Security Newsgroups

Security General at http://www.microsoft.com/technet/community/newsgroups/dgbrowser/en-us/default.mspx?dg=microsoft.public.security&lang=en&cr=US and news://msnews.microsoft.com/microsoft.public.security

Security HfNetChk at http://www.microsoft.com/technet/community/newsgroups/dgbrowser/en-us/default.mspx?dg=microsoft.public.security.hfnetchk&lang=en&cr=US and news://msnews.microsoft.com/microsoft.public.security.hfnetchk

Security MBSA at http://www.microsoft.com/technet/community/newsgroups/dgbrowser/en-us/default.mspx?dg=microsoft.public.security.baseline_analyzer&lang=en&cr=US and news://msnews.microsoft.com/microsoft.public.security.baseline_analyzer

Security Toolkit at http://www.microsoft.com/technet/community/newsgroups/dgbrowser/en-us/default.mspx?dg=microsoft.public.security.toolkit&lang=en&cr=US and news://msnews.microsoft.com/microsoft.public.security.toolkit

Security Virus at http://www.microsoft.com/technet/community/newsgroups/dgbrowser/en-us/default.mspx?dg=microsoft.public.security.virus&lang=en&cr=US and news://msnews.microsoft.com/microsoft.public.security.virus

Summary

Before you can dive into a patch management deployment process, you must establish the prerequisites for implementing the process by knowing your computing environment, preparing end user education, assigning responsibilities, understanding the current process, developing a chain of communication, baselining, and acquiring management buy-off. You are then ready to undertake the four phases of the patch management process—assess, identify, evaluate and plan, and deploy:

- In the assessment phase, determine
    - What you have in your production environment
    - What security threats and vulnerabilities you might face
    - Whether your organization is prepared to respond to new software updates
- In the identification phase, your goals are to
    - Discover new software updates in a reliable way
    - Determine whether an update is relevant to your production environment
    - Determine the deployment classification for an update

- In the evaluation and planning phase, your goals are to
    - Make a go/no-go decision to deploy a software update
    - Determine what is needed to deploy an update
    - Test the software update in a production-like environment to confirm that it does not compromise business-critical systems and applications
    - Gain approval for deployment
    - Pass the tested updated to the deployment team
- In the deploy phase, your goals are to
    - Successfully roll out the approved software update into your production environment
    - Meet all of the requirements of any deployment SLAs
    - Update baselines in preparation for the next process cycle

In the next chapter, we'll look at patch management tools and how you can evaluate the available technologies to determine which option is right for your environment.