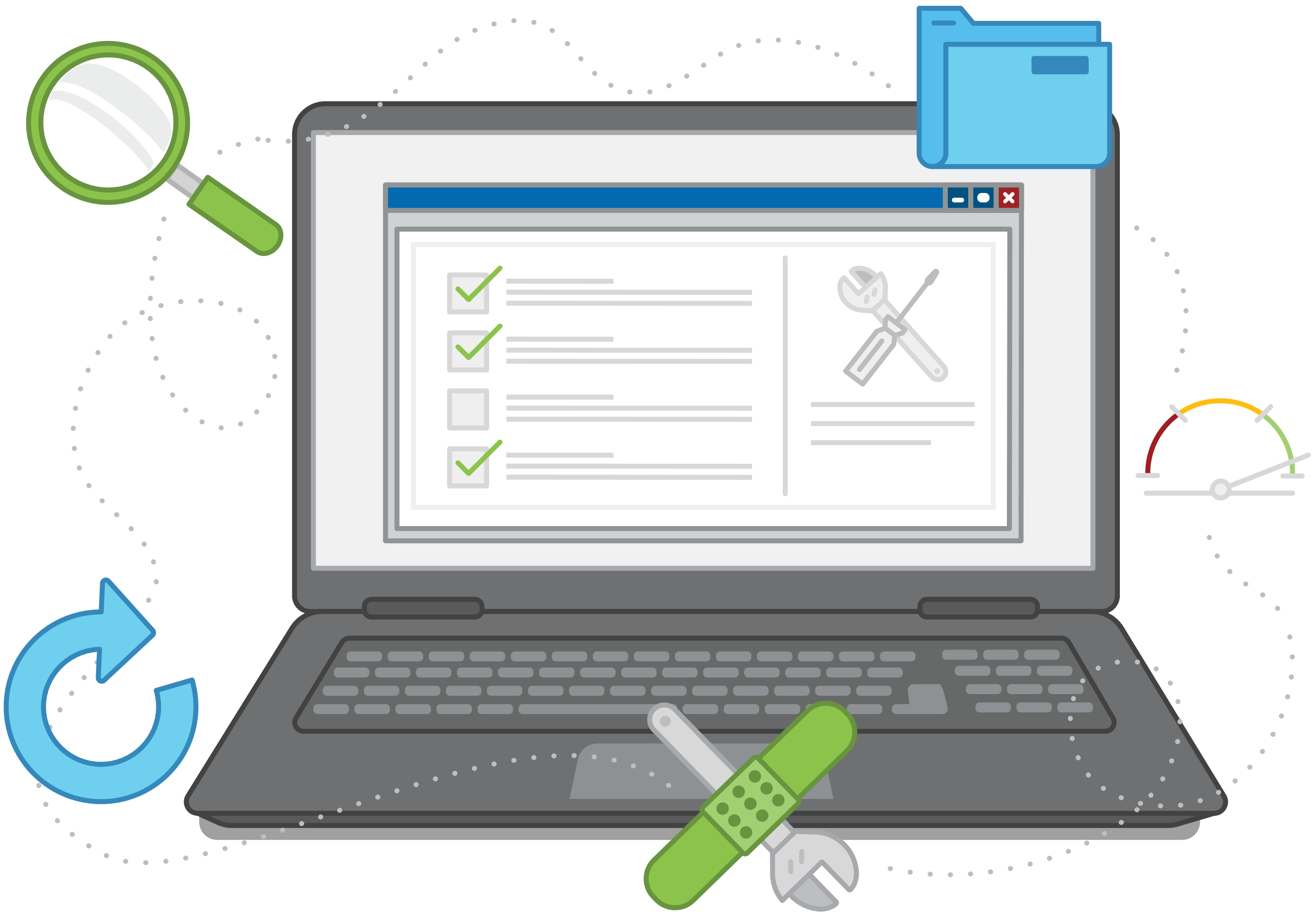
ConnectWise®

ConnectWise eBook Series



Patch Management

Best Practices

Contents

Introduction

What is Patch Management?

Patch management is the practice of reviewing, understanding, testing, deploying, and reconciling the deployment state for software product updates. The goal of the updates is to correct problems, close vulnerabilities, and improve product functionality, which is essential to the stability of an IT infrastructure in most environments. By understanding the different kinds of patches and following best practices, an IT service provider can keep clients' critical systems free from known vulnerabilities.

Patch management is probably the biggest concern of IT service providers and their clients these days.

With new vulnerabilities being discovered almost daily, keeping systems up-to-date with patches is often a full-time job, especially in larger environments. In addition, the lag time between when a vulnerability is discovered and when a virus or worm appears is now measured in days or weeks rather than months. This puts tremendous pressure on vendors to release patches before they've been fully regression-tested. The result is that oftentimes patches fix the problem they're designed to address, but unintentionally break something else in the process. Most IT service providers pay attention to security and patching their clients' systems, but how many have a well-honed patch management policy? Patch management is often seen as a trivial task. Simply click on 'update' and that's it. But in reality, there is a lot more to it and a proper policy is certainly not overkill. But what should a patch management policy include apart from deploying patches?

Read on to learn how to implement patch management policies, processes, and persistence. Plus, gain valuable patching resources and tools.

Types of Patches

Before you plan a patch management strategy, it's important to understand the differences between the various flavors of patches. Microsoft® classifies patches into three basic categories: hotfixes, roll-ups, and service packs.

Hotfixes

Hotfixes are small patches designed to fix a single problem and are developed either in response to a security advisory or by customer request. Hotfixes are typically issued to either plug security holes, such as buffer overflows, or to fix features that don't behave as intended.

Roll-Ups

Occasionally, Microsoft combines several hotfixes together into a single package called a roll-up. This is typically done when several security issues have been identified within a short time period, and its purpose is to simplify the job of installing hotfixes for administrators. Unfortunately, this is not always a good idea. There have been instances in which installing multiple patches broke applications, and the headache then arises – figuring out which patch in the roll-up actually caused the problem.

Service Packs

At fairly regular intervals, Microsoft combines all hotfixes issued for a platform into a single package called a service pack. These service packs are cumulative. For instance, Service Pack 3 includes all hotfixes issued both before and since Service Pack 2 appeared. While service packs undergo more thorough testing than individual hotfixes, there have nevertheless been a few instances in which a service pack caused new problems while solving others.

MSRC Ratings System

Hotfixes that address security vulnerabilities are also called security fixes and the Microsoft Security Resource Center (MSRC) rates these according to a four-point scale from high to low. This is useful for administrators because it allows them to decide which fixes should be applied as soon as possible and which can be deferred until later or even ignored. The ratings also refer to the types of vulnerabilities they guard against. An example of a critical issue might be a self-propagating Internet worm that can bring servers to their knees and wreak other kinds of havoc, which means that your clients' confidential business information might be at risk of being lost, stolen or corrupted. Moderate means you have a properly configured firewall and are following good security practices, so you aren't likely to be affected by the problem, although it's still possible. Finally, low means it would take a combination of a genius hacker and a totally negligent system administrator for the exploit to occur, but it's still remotely possible.

Patch Management Best Practices:

Policy, Process, and Persistence

Effective patch management can be summarized as **policy, process,** and **persistence** (PPP). The following pages unravel these areas and provide some helpful recommendations from Microsoft®.

Policy

The first step in developing a patch management strategy is to develop a policy that outlines the who, what, how, when and why of patching your clients' systems. **This advance planning enables you to be proactive instead of reactive.** Proactive management anticipates problems in advance and develops policies to deal with them; reactive management adds layer upon layer of hastily thought-up solutions patched together using bits of string and glue. It's easy to see which approach will unravel in the event of a crisis.

When you have a patch management policy in place, and a notification arrives of a critical vulnerability in a software product, you immediately know who will deal with it, how you will deploy the patch, whether it needs to be done sooner or later, and so on. For example, a simple element of a patch management policy might be that critical or important patches should be applied immediately, while moderate or low patches should be submitted to a team member for further study. Another example is proactively scheduling a specific day of the week or month for installing patches (usually weekends, in case something breaks), as opposed to the drop-everything, the-sky-is-falling approach common in a reactive environment. Making a decision tree that addresses these issues ahead of time reduces anxiety and speeds response when the time comes to patch something.

Process

The detailed procedure you will use to respond to vulnerabilities and deploy patches should be explicit within your security policy. The typical patch management process is illustrated above by the process workflow in general terms, and includes aspects of the Information Technology Infrastructure Library (ITIL) to ensure success.

The following six-step process is defined as best practice by Microsoft and should also be considered as you craft your own tailor-made process for use within your managed services practice.

1 **Notification.** Information comes to you about a vulnerability with a patch meant to eliminate it. Notification might be sent via email from the Microsoft Security Notification Service, a pop-up balloon when you're using Automatic Updates, a message displayed in the Software Update Services (SUS) web console, or some other method. It all depends on which tools you use to keep your systems patched and up-to-date.

2 **Assesment.** Based on the patch rating and the configuration of your systems, you need to decide which systems need the patch, and how quickly they need to be patched to prevent an exploit. Having an accurate inventory of systems and applications running on your clients' networks is essential if you want to keep the networks secure against intrusion.

3 **Obtainment.** How you get the patch you need depends on which patch management tools you choose to deploy. In general, such tools range from completely manual (i.e. visiting the Windows Update website) to almost entirely automatic (i.e. via remote monitoring and management software).

Process Page 2 Continued

4 **Testing.** Information Testing should always take place before you apply patches to production systems. Test your patches on a test bed network that simulates your production network. Remember that Microsoft can't test all possible effects of a patch before releasing it, because there are thousands of applications that can run on servers and millions of combinations of applications. Thus, you must test patches before deploying them, especially if you have custom code running on your machines. If you need a way to justify the cost of purchasing duplicate equipment for a test bed network, tell the boss it's like insurance. If you deploy patches to a client that has 15 systems and you wreck all of them at the same time, that client is effectively out of business until you get everything restored. If you can't afford to lose a client, you need to plan for some level of patch testing.

5 **Deployment.** Based Deploy a patch only after you've thoroughly tested it. When you are ready to apply it, do so carefully. Don't apply a patch to all your systems at once, just in case your testing process missed something. A good approach is to apply patches one at a time, testing your production servers after each patch is applied to make sure applications still function properly. A major consideration to deploying should also be based on geographic location. If you have a client with three locations, you should consider applying patches on three separate days to avoid a situation where you potentially take out the entire company if one patch has an issue following deployment. It is certainly better to be safe than sorry in this case, and the little extra care will go a long way with client relations if something negative were to result from the patch cycle.

6 **Validation.** How The final step in the process is often forgotten: making sure that the patch has actually been installed on the targeted systems. The validation process must be completed so when it comes time to report on status to your client, you are certain that the data being submitted is an accurate representation of the actual patch status. This reporting and validation process takes some time, but it is a necessary procedure to ensure that service levels are met.

Persistence

Policies are useless and processes are futile unless you persist in applying them consistently. Network security requires constant vigilance, not only because new vulnerabilities and patches appear almost daily, but because new processes and tools are constantly being developed to handle the growing problem of keeping systems patched.

Effective patch management has become a necessity in today's information technology environments. Reasons for this necessity are:

- The ongoing discovery of vulnerabilities in existing operating systems and applications
- The continuing threat of attackers developing applications that exploit those vulnerabilities
- Vendor requirements to patch vulnerabilities via the release of patches and updates

These points illustrate the need to constantly apply patches to your clients' IT environments. **Such a large task is best accomplished following a series of repeatable, automated best practices.** Therefore, it's important to look at patch management as a closed-loop process. It is a series of best practices that have to be repeated regularly on your clients' networks to ensure protection from exposed vulnerabilities. Patch management requires the regular rediscovery of systems that may potentially be affected, scanning those systems for vulnerabilities, downloading patches and patch definition databases, and deploying patches to systems that need them.

Patch Management requires:

1. Regular rediscovery of systems that may potentially be affected
2. Scanning those systems for vulnerabilities
3. Downloading patches and patch definition databases
4. Deploying patches to systems that need them

Patching Resources

Microsoft updates arrive predictably on Patch Tuesday (the second Tuesday of every month), which means you can plan ahead for testing and deployment. You can get advance notice by subscribing to the security bulletin, which comes out three business days before the release and includes details of the updates.

The following is a list of currently available resources you can use when augmenting your patch process, as well as some that can keep you informed of patch-related updates that fall outside the scope of Microsoft updates.

Microsoft Security TechCenter >>

SearchSecurity Patch News >>

Oracle Critical Patch Updates and Security Alerts >>

PatchManagement.org (Patch Mailing List) >>

Patch My PC (third-party, free patching) >>

Remote Monitoring and Management

Approving and deploying patches on individual machines is not scalable. As your business grows and you take on more clients, it is important to utilize a tool that can automate your patch management process so your technicians aren't bogged down with the mundane task of individually patching each client machine. A remote monitoring and management (RMM) platform with built-in patch management capabilities can help.

An RMM platform is typically used by IT service providers to remotely monitor and manage their clients' IT systems from a centralized console. However, some RMM tools go a step further and enable IT service providers to automate certain maintenance tasks, such as patch management. When looking for an RMM platform with patch management capabilities, look for one that enables you to:

6 Patching Tools

Identify, approve, update, or ignore patches and hotfixes for one or multiple devices at a group level

Define patch install windows for an individual device or a group of devices

Schedule patch installation times and patch reboot times

Create tickets for all successful patch install jobs

Provide detailed reports of patch install jobs to your management team

Third Party Patch Management

It is important to ensure timely installation of patches so security holes remain closed not only in the Windows operating system, but also in other applications that are used on desktops and servers. Third Party Patch Management natively extends ConnectWise Automate so that you can begin auditing, patching, documenting, and even billing for third party application updates.

Summary

Patch management is a fundamental service offered in most managed service provider (MSP) service plans and is a critical process in protecting your clients' systems from known vulnerabilities and potential exploits that could result in their systems being compromised. Viruses and malware are just one example of aggressors that take advantage of these vulnerabilities and can be especially destructive and difficult to correct. Clients that are impacted by something that could and should have been prevented by utilizing a comprehensive patch policy are likely to begin looking for alternative service providers.

Patches correct bugs, flaws and provide enhancements, which can prevent potential user impact, improve user experience and potentially save your technicians time researching and repairing issues that could have already been resolved or prevented with an existing update. Clients generally understand that their systems need to be patched, but they likely do not have the expertise to comfortably approve and install patches without help. When operating within the IT services industry, patching is one of the first areas that competitors auditing your clients will assess.

Developing best practices to manage the risks associated with the approval and deployment of patches is critical to your IT department's service offering.

ConnectWise®

About ConnectWise

ConnectWise transforms how Technology Teams successfully build, manage, and grow their businesses. Our award-winning set of software solutions provide a seamless experience to companies in more than 70 countries, giving them the ability to increase their productivity, efficiency, and profitability. When combined with our passion, commitment to innovation, and more than 35 years of experience, ConnectWise software solutions deliver the results companies want at each step of their business journey. Attend one of our events and see the benefits of our game-changing community firsthand.

For more information, visit ConnectWise.com.

PatchManagementBestPractices-US-0718