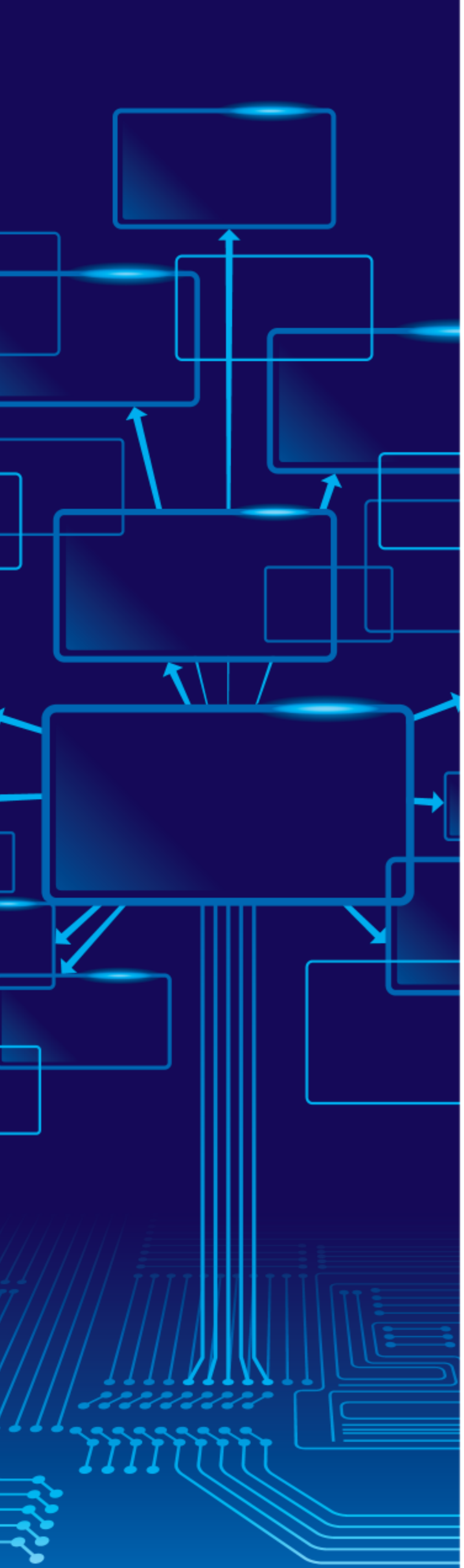


What is patch management? Lifecycle, benefits and best practices

In this guide:

This comprehensive guide explains the entire patch management process and its role in IT administration and security. The hyperlinks direct you to detailed articles on patch management best practices, tools and services.

In this guide:

What is patch management? Lifecycle, benefits and best practices

DAVID ESSEX, INDUSTRY EDITOR

Patch management is the subset of systems management that involves identifying, acquiring, testing and installing patches, or code changes, that are intended to fix bugs, close security holes or add features.

Patch management requires staying current on available patches, deciding which patches are needed for specific software and devices, testing them, making sure they have been properly installed and documenting the process.

This comprehensive guide explains the entire patch management process and its role in IT administration and security. The hyperlinks direct you to detailed articles on patch management best practices, tools and services.

WHY IS PATCH MANAGEMENT IMPORTANT?

Patch management helps keep computers and networks secure, reliable and up to date with features and functionality that the organization considers important. It is also an essential tool for ensuring and documenting compliance with security and privacy regulations. Patching can improve performance and is sometimes used to bring software up to date, so it will work with the latest hardware.

HOW DOES PATCH MANAGEMENT WORK?

Patch management works differently depending on whether a patch is being applied to a standalone system or systems on a corporate network. On a standalone system, the operating system and applications will periodically perform automatic checks to see if patches are available. New patches will typically be downloaded and installed automatically.

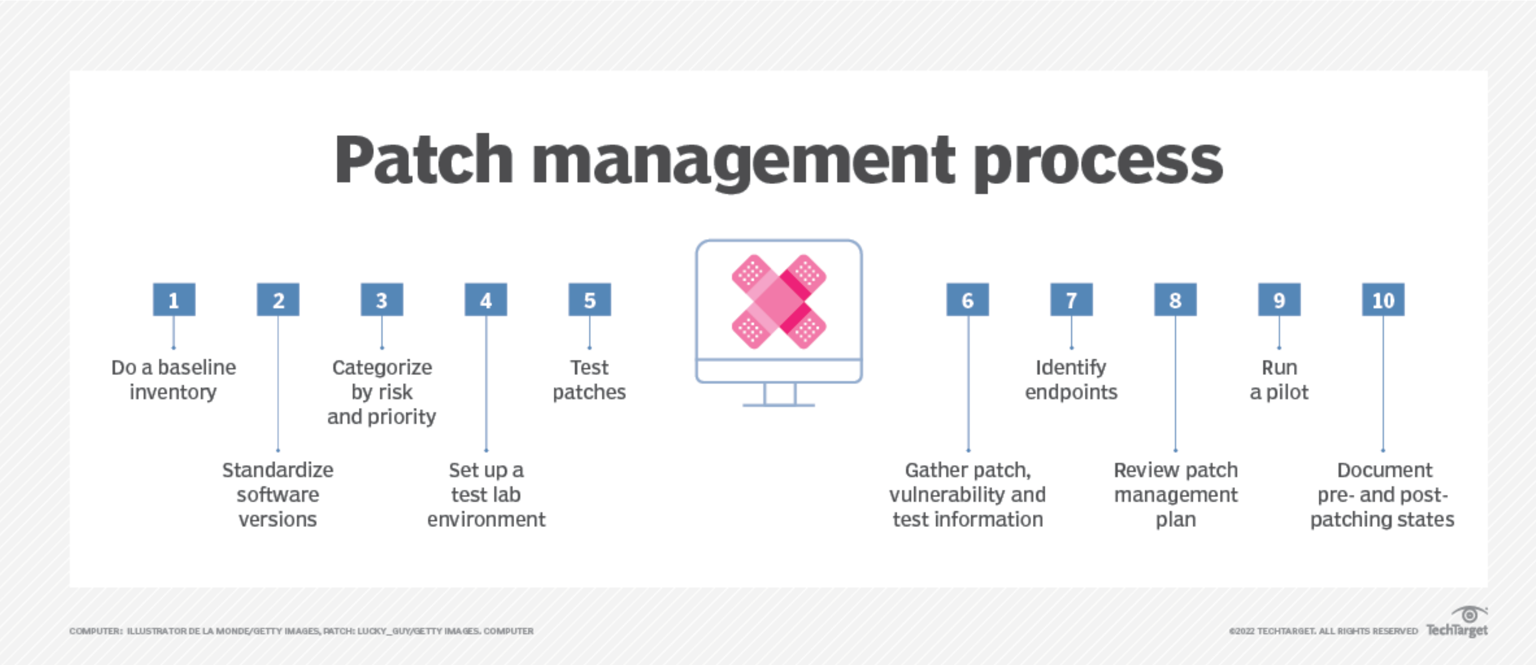
In networked environments, organizations generally try to maintain software version consistency across computers and usually perform centralized patch management rather than allowing each computer to download its own patches. Centralized patch management uses a central server that checks network hardware for missing patches, downloads the missing patches and distributes them to the computers and other devices on the network in accordance with the organization's patch management policy.

A centralized patch management server does more than just automate patch management; it also gives the organization a degree of control over the patch management process. For example, if a particular patch is determined to be problematic, the organization can configure its patch management software to prevent the patch from being deployed.

Another advantage of centralized patch management is that it helps conserve internet bandwidth. It makes little sense from a bandwidth perspective to allow every computer in an organization to download the exact same patch. Instead, the patch management server can download the patch once and distribute it to all the computers designated to receive it.

Although many organizations handle patch management on their own, some managed service providers perform patch management in conjunction with the other network management services they provide to clients. MSP patch management can minimize the significant administrative hassles of doing the work in-house.

In this guide:


Patch management process
1 Do a baseline inventory
2 Standardize software versions
3 Categorize by risk and priority
4 Set up a test lab environment
5 Test patches
6 Gather patch, vulnerability and test information
7 Identify endpoints
8 Review patch management plan
9 Run a pilot
10 Document pre- and post-patching states

COMPUTER: ILLUSTRATOR DE LA MONDE/GETTY IMAGES, PATCH: LUCKY_GUY/GETTY IMAGES. COMPUTER

©2022 TECHTARGET. ALL RIGHTS RESERVED TechTarget

WHAT ARE THE BENEFITS OF PATCH MANAGEMENT?

Most major software companies periodically release patches, which can serve any of three primary purposes:

1. Patches are often used to address security vulnerabilities. If a software vendor discovers a security risk associated with one of its products, it will typically issue a patch intended to address that risk. It is important for organizations to apply security patches as soon as

possible because hackers and malware authors know about the security vulnerabilities a patch is designed to correct and will actively look for unpatched systems.

2. Patches can fix bugs, improving the software's stability and eliminating annoying problems.
3. Vendors occasionally release patches to introduce new features. Feature updates are becoming more common because of the growth of subscription-based cloud software.

WHAT ARE THE CHALLENGES OF PATCH MANAGEMENT?

Buggy patches are the most common problem in patch management. Sometimes a patch will introduce problems that did not exist before. They may show up in the product that is being patched or in other software that has a dependency relationship with the patched software. A patch might also have to be removed if the vendor releases a patch that can't be put in place while the previous patch remains on the system. Because patches can sometimes introduce problems into a system that was previously working correctly, it is important for administrators to test patches before deploying them.

Another common problem is that disconnected systems might not receive patches in a timely manner. For example, if a mobile user rarely connects to the corporate network, their device may go for long periods without being patched. In such cases, it may be better to configure the device for standalone patch management rather than relying on centralized patch management.

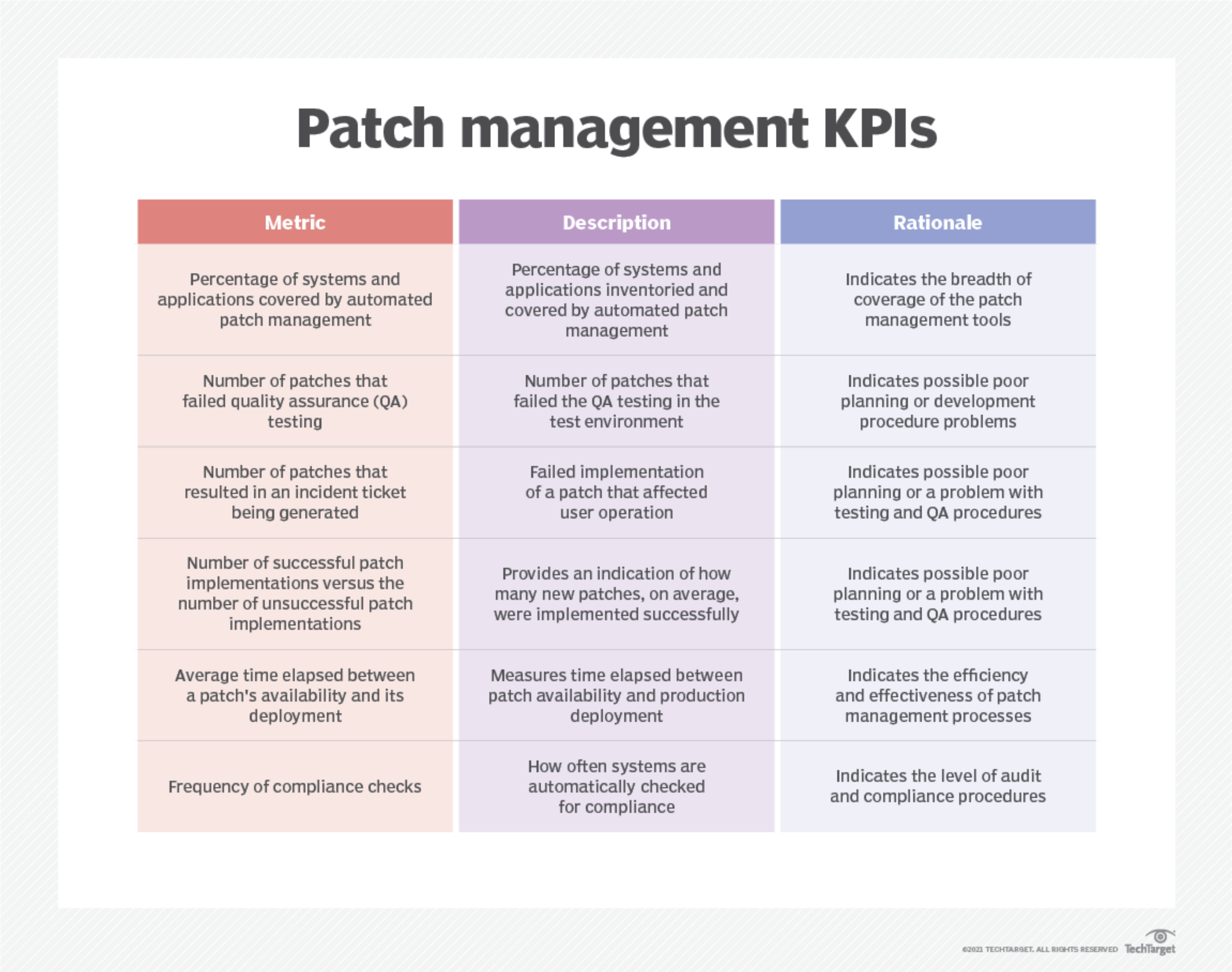
The sharp increase in remote work since the start of the COVID-19 pandemic has added a new problem: managing patches on a wider range of endpoints that connect to the network through various security mechanisms. While some users might connect to applications on a highly secure VPN, others might use single sign-on from the public internet, log into some applications individually or use unsecure Wi-Fi networks. There are more places for hackers to enter the corporate network, which can mean more patches to deploy.

PATCH MANAGEMENT LIFECYCLE

The main stages of the patch management process -- identifying, acquiring, testing, deploying and documenting them -- are supported by the following important steps:

- inventorying devices, operating systems and applications;
- deciding which software versions to standardize on;
- categorizing IT assets and patches by risk and priority;
- testing patches in a representative lab or sandbox environment;
- running a pilot on a sample of devices (an optional step);
- validating patches to confirm that they have been installed and to detect systems that are missing patches;
- planning the rollout, including identifying who is responsible for it and which patches should be installed on which devices; and
- documenting patches, vulnerabilities, test results and deployments, which helps in analyzing and improving the process.

In this guide:



Patch management KPIs

Metric	Description	Rationale
Percentage of systems and applications covered by automated patch management	Percentage of systems and applications inventoried and covered by automated patch management	Indicates the breadth of coverage of the patch management tools
Number of patches that failed quality assurance (QA) testing	Number of patches that failed the QA testing in the test environment	Indicates possible poor planning or development procedure problems
Number of patches that resulted in an incident ticket being generated	Failed implementation of a patch that affected user operation	Indicates possible poor planning or a problem with testing and QA procedures
Number of successful patch implementations versus the number of unsuccessful patch implementations	Provides an indication of how many new patches, on average, were implemented successfully	Indicates possible poor planning or a problem with testing and QA procedures
Average time elapsed between a patch's availability and its deployment	Measures time elapsed between patch availability and production deployment	Indicates the efficiency and effectiveness of patch management processes
Frequency of compliance checks	How often systems are automatically checked for compliance	Indicates the level of audit and compliance procedures

PATCH MANAGEMENT BEST PRACTICES

System management software vendors, MSPs and consultants have expertise in making patch deployment smooth and effective. Among the oft-mentioned patch management best practices are the following 10 recommendations:

1. **Know what you're responsible for patching.** This entails clearly identifying targets and their locations.
2. **Create standard and emergency patching procedures.** Emergency patches have to be installed outside the windows established for regularly scheduled patching. There should be clear procedures for both.
3. **Understand vendor patch-release schedules.** The number and types of operating systems, applications and endpoint firmware vary significantly, as does the timing of the patches available for them.
4. **Design and maintain a realistic test environment.** It should closely match the production environment, including workload fluctuations, and will need to be updated when the production environment is changed. This can be expensive and hard to scale, so a representative sample of assets is often used instead. Another option is a virtual test environment designed to replicate the production environment on a single computer or cloud service such as AWS or Microsoft Azure. There are also online services that handle the replication process.
5. **Review the patch process and results.** Your review of patch management KPIs can also help identify potential improvements.

6. **Prioritize patches by risk level.** Assign assets a criticality level according to their importance to business processes, optimal downtime and vulnerability risk. Test, schedule and deploy patches for the most critical assets before less essential ones.
7. **Stay abreast of security vulnerabilities.** For commercial software, subscribe to reputable sources, such as the Cybersecurity and Infrastructure Security Agency or Common Vulnerability Scoring System of the U.S. government. For internally developed applications, use a software composition analysis tool to track their open source and third-party components.
8. **Deploy patches as quickly as possible.** Users may complain about downtime, but the longer the wait, the greater the chance that hackers will find a way in.
9. **Execute production rollouts in stages.** Dedicate the initial rollout to less critical systems. If the patches perform as expected, continue the rollout until every system is updated.
10. **Create a contingency and rollback plan.** In the event something goes wrong, have a backup or image snapshot of systems before starting patch deployment so they can be restored to their previous state.

EXAMPLES OF PATCH MANAGEMENT

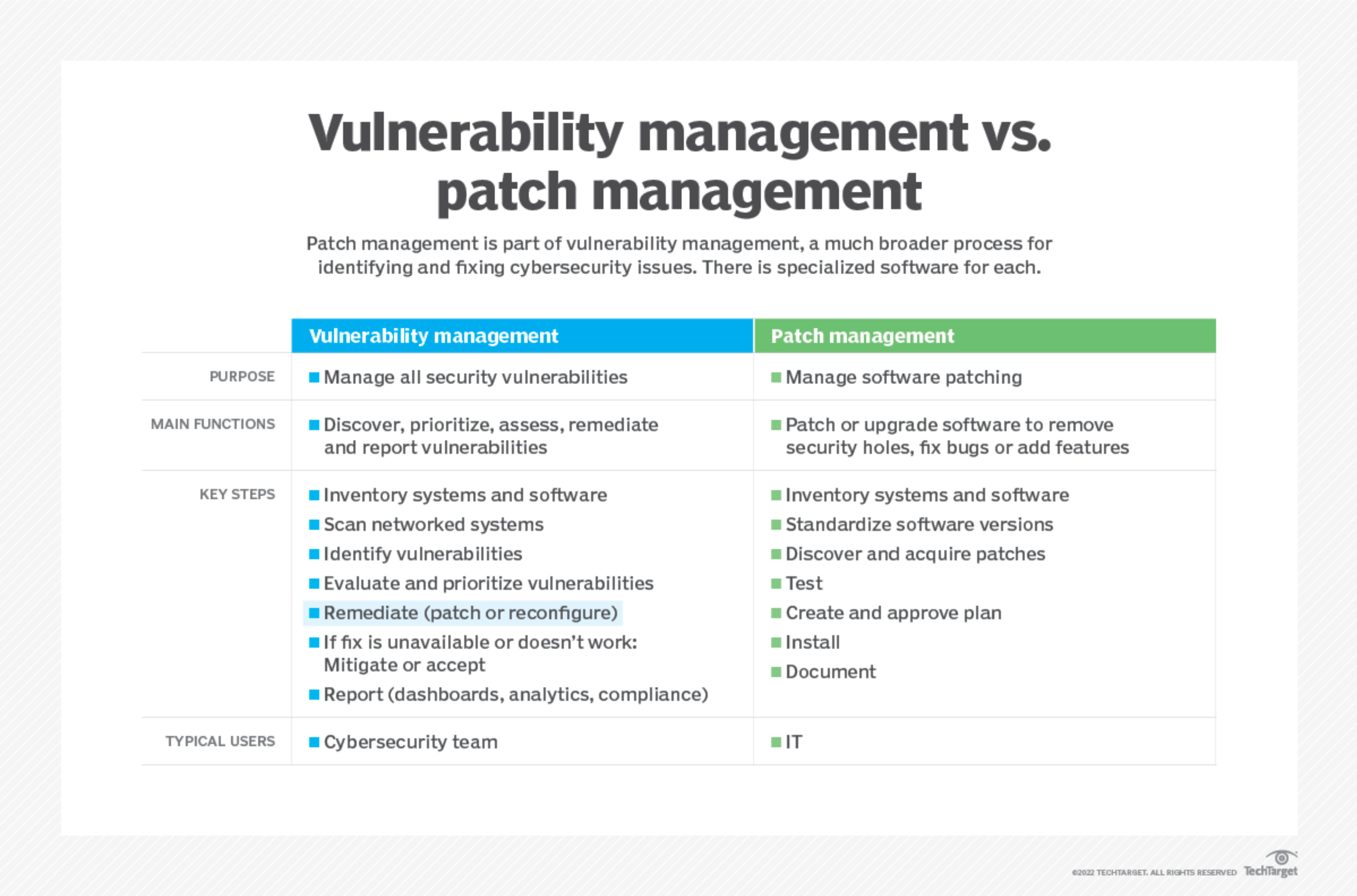
Microsoft often provides patches to its Windows operating systems and other products such as Office. The patches are normally released on a scheduled monthly basis, often on a day that has come to be known as Patch Tuesday.

Standalone systems rely on Windows Update to automatically download and deploy any available patches. In business environments, however, it is much more common to use Windows Server Update Services (WSUS), which are included with Windows Server and specifically designed to centralize patch management. There are also numerous third-party WSUS alternatives for managing, downloading and deploying Microsoft patches.

Many IT departments also maintain systems that run the open source Linux operating system. Linux patch management is similar to Windows patching, but there are more Linux distributions, which means becoming familiar with the different patching procedures of several vendors instead of just one.

MacOS also has built-in software update tools, but an organization can have multiple versions of the operating system, which makes it challenging to keep every system up to date without using centralized patch management. Many third-party patch management tools support macOS, along with Windows and Linux.

In this guide:



Vulnerability management vs. patch management

Patch management is part of vulnerability management, a much broader process for identifying and fixing cybersecurity issues. There is specialized software for each.

	Vulnerability management	Patch management
PURPOSE	■ Manage all security vulnerabilities	■ Manage software patching
MAIN FUNCTIONS	■ Discover, prioritize, assess, remediate and report vulnerabilities	■ Patch or upgrade software to remove security holes, fix bugs or add features
KEY STEPS	■ Inventory systems and software ■ Scan networked systems ■ Identify vulnerabilities ■ Evaluate and prioritize vulnerabilities ■ Remediate (patch or reconfigure) ■ If fix is unavailable or doesn't work: Mitigate or accept ■ Report (dashboards, analytics, compliance)	■ Inventory systems and software ■ Standardize software versions ■ Discover and acquire patches ■ Test ■ Create and approve plan ■ Install ■ Document
TYPICAL USERS	■ Cybersecurity team	■ IT

PATCH MANAGEMENT IN CYBERSECURITY AND VULNERABILITY MANAGEMENT

The increase in cyber attacks in recent years makes cybersecurity probably the most common reason for deploying patches. Patch management is an important part of *vulnerability management*, a much broader strategy for discovering, prioritizing and remediating the security vulnerabilities of network assets. Patch management remediates the identified risks by upgrading software to the most recent version or by temporarily patching it to remove a vulnerability until the software vendor releases an upgrade that contains the fix.

In this context, software patch testing also involves documenting the test process for security compliance purposes, as well as coming up with alternative vulnerability management plans in case security patches can't be installed on the required devices.

Vulnerability management includes the following steps:

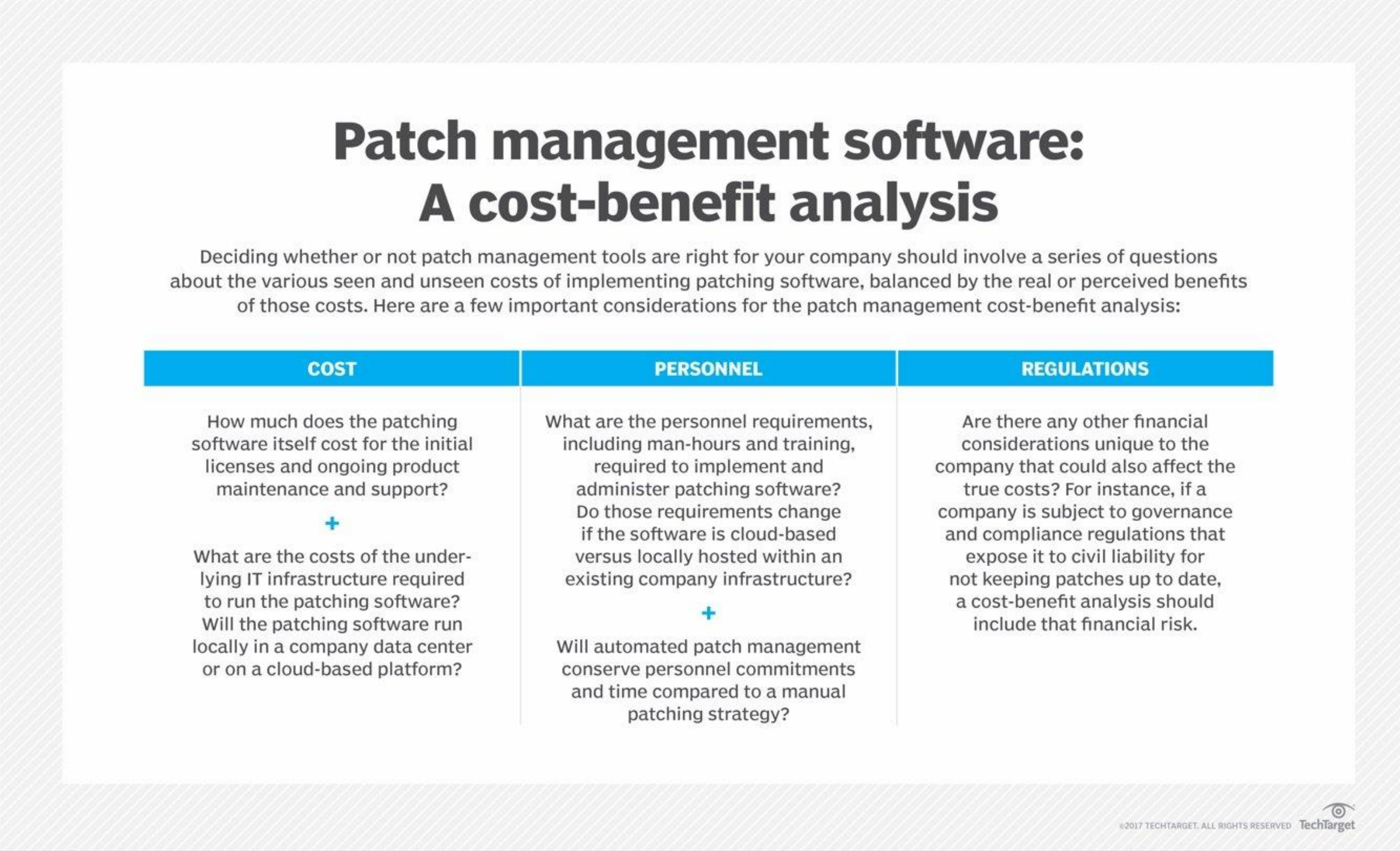
- network scanning to identify users and devices on the network (the same technique hackers use to search for vulnerable targets);
- penetration testing, which mimics the tactics of hackers to identify vulnerable parts of the network;
- verification to confirm that a vulnerability identified during scanning and testing can, in fact, be exploited;

- mitigation steps, such as taking a vulnerable system offline, to prevent vulnerabilities from being exploited before a patch is available; and
- reporting that uses data management, analytics and visualization tools for evaluating the organization's vulnerability management process and complying with regulations.

A distinct category of tools known as *vulnerability management software* is used for scheduling and documenting these processes and partly automating them. Some vulnerability management tools have patch management as a component.

HOW TO CHOOSE THE RIGHT PATCH MANAGEMENT SOFTWARE

Patch management tools are available on premises or in the cloud, and many vendors offer both deployment options. While some vendors specialize in patch management, most include it in a broader collection of IT systems management, endpoint management or security and compliance tools. Prominent players include Atera, Automox, GFI LanGuard, Kaseya VSA, ManageEngine Patch Manager Plus and SolarWinds Patch Manager.



Patch management software: A cost-benefit analysis

Deciding whether or not patch management tools are right for your company should involve a series of questions about the various seen and unseen costs of implementing patching software, balanced by the real or perceived benefits of those costs. Here are a few important considerations for the patch management cost-benefit analysis:

COST	PERSONNEL	REGULATIONS
How much does the patching software itself cost for the initial licenses and ongoing product maintenance and support? + What are the costs of the underlying IT infrastructure required to run the patching software? Will the patching software run locally in a company data center or on a cloud-based platform?	What are the personnel requirements, including man-hours and training, required to implement and administer patching software? Do those requirements change if the software is cloud-based versus locally hosted within an existing company infrastructure? + Will automated patch management conserve personnel commitments and time compared to a manual patching strategy?	Are there any other financial considerations unique to the company that could also affect the true costs? For instance, if a company is subject to governance and compliance regulations that expose it to civil liability for not keeping patches up to date, a cost-benefit analysis should include that financial risk.

Before investigating products, it's important to create a complete patch management policy. By ranking their reasons for deploying patches and specifying who needs to be involved, how patches will be tested, implemented and monitored, and what kind of reporting is required, organizations will be more successful at finding the software that meets their exact needs.

Buying teams should look for dashboards that are easy to set up, understand and use, and that can display the information they need. Reporting and documentation features should also be user-friendly and able to handle the required information on vulnerabilities, test results and patching history. The software should support patching for every operating system and major application used in the organization. Most vendors name the OSes and commercial applications their products can patch.

Other important patch management features include the following:

- real-time visibility into network devices and software;
- the ability to prioritize patches by criticality;
- administrative controls for approving patches;
- patch automation -- sometimes through the use of software agents -- with fine tuning that makes it easy to exclude certain patches and endpoints;
- timely receipt of software updates and patches from vendors;
- the ability to program patch management policies into the system; and
- patch validation and verification to confirm that patches have taken effect.

Organizations that take the time to develop a patch management policy, establish a comprehensive patch management process and use the software tools that best support those efforts will likely be successful at making their IT systems reliable, secure and current with the latest technology.

▼ CONTINUE READING

- Use this 10-step patch management process to ensure success
- The risks of failed patch management
- Creating a patch management policy: Step-by-step guide