

LEARNING MADE EASY

NinjaOne Special Edition

# Patch Management

for  
**dummies**<sup>®</sup>  
A Wiley Brand



Address and  
conquer patch challenges

Automate patch  
management

Maintain better  
security

Brought to you  
by

**ninjaOne**<sup>®</sup>

Kenneth Hess

# About NinjaOne

NinjaOne automates the hardest parts of IT, empowering more than 17,000 IT teams with visibility, security, and control over all endpoints. The NinjaOne platform is proven to increase productivity, while reducing risk and IT costs. NinjaOne is consistently ranked #1 for its world-class support and is the top-rated software on G2 in seven categories including endpoint management, remote monitoring and management, and patch management. Try NinjaOne for free at [www.ninjaone.com/freetrialform](https://www.ninjaone.com/freetrialform).



# Patch Management

NinjaOne Special Edition

**by Kenneth Hess**

for  
**dummies**<sup>®</sup>  
A Wiley Brand

# Patch Management For Dummies®, NinjaOne Special Edition

Published by  
**John Wiley & Sons, Inc.**  
111 River St.  
Hoboken, NJ 07030-5774  
\*\*\*.wiley.com

Copyright © 2024 by John Wiley & Sons, Inc., Hoboken, New Jersey. All rights, including for text and data mining, AI training, and similar technologies, are reserved.

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without the prior written permission of the Publisher. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

**Trademarks:** Wiley, For Dummies, the Dummies Man logo, The Dummies Way, Dummies.com, Making Everything Easier, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates in the United States and other countries, and may not be used without written permission. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc., is not associated with any product or vendor mentioned in this book.

LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: THE PUBLISHER AND THE AUTHOR MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS WORK AND SPECIFICALLY DISCLAIM ALL WARRANTIES, INCLUDING WITHOUT LIMITATION WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES OR PROMOTIONAL MATERIALS. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR EVERY SITUATION. THIS WORK IS SOLD WITH THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING LEGAL, ACCOUNTING, OR OTHER PROFESSIONAL SERVICES. IF PROFESSIONAL ASSISTANCE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL PERSON SHOULD BE SOUGHT. NEITHER THE PUBLISHER NOR THE AUTHOR SHALL BE LIABLE FOR DAMAGES ARISING HEREFROM. THE FACT THAT AN ORGANIZATION OR WEBSITE IS REFERRED TO IN THIS WORK AS A CITATION AND/OR A POTENTIAL SOURCE OF FURTHER INFORMATION DOES NOT MEAN THAT THE AUTHOR OR THE PUBLISHER ENDORSES THE INFORMATION THE ORGANIZATION OR WEBSITE MAY PROVIDE OR RECOMMENDATIONS IT MAY MAKE. FURTHER, READERS SHOULD BE AWARE THAT INTERNET WEBSITES LISTED IN THIS WORK MAY HAVE CHANGED OR DISAPPEARED BETWEEN WHEN THIS WORK WAS WRITTEN AND WHEN IT IS READ.

For general information on our other products and services, or how to create a custom *For Dummies* book for your business or organization, please contact our Business Development Department in the U.S. at 877-409-4177, contact [info@dummies.biz](mailto:info@dummies.biz), or visit [www.dummies.com/custom-solutions](http://www.dummies.com/custom-solutions). For information about licensing the *For Dummies* brand for products or services, contact [BrandedRights&Licenses@wiley.com](mailto:BrandedRights&Licenses@wiley.com).

ISBN 978-1-394-27006-4 (pbk); ISBN 978-1-394-27007-1 (ebk); ISBN 978-1-394-28841-0 (epub)

## Publisher's Acknowledgments

**Editor:** Elizabeth Kuball

**Acquisitions Editor:** Traci Martin

**Senior Managing Editor:**  
Rev Mengle

**Client Account Manager:**  
Cynthia Tweed

**Production Editor:**  
Tamilmani Varadharaj

# Table of Contents

INTRODUCTION .....	1
About This Book .....	1
Foolish Assumptions.....	1
Icons Used in This Book.....	2
Beyond the Book.....	2
<b>CHAPTER 1: Introducing Patch Management.....</b>	<b>3</b>
Realizing the Criticality of Patch Management .....	3
Defining Patch Management .....	4
Identifying Patch Types.....	5
Outlining the Patch Management Life Cycle.....	5
Step 1: Identifying assets .....	6
Step 2: Prioritizing patches .....	6
Step 3: Creating policies.....	6
Step 4: Monitoring patch status.....	6
Step 5: Testing patches .....	6
Step 6: Documenting changes.....	6
Step 7: Deploying patches .....	7
Step 8: Auditing post-patch issues.....	7
Step 9: Reporting compliance .....	7
Step 10: Reviewing, updating, and repeating steps.....	7
Handling the Challenges.....	7
<b>CHAPTER 2: Implementing a Patch Management Strategy ....</b>	<b>9</b>
Prioritizing and Assessing Risks.....	9
Developing a Policy .....	10
Testing and Validating Patches.....	10
Monitoring the Process .....	12
Reporting on Patch Compliance and Effectiveness .....	12
<b>CHAPTER 3: Exploring Patch Management Tools and Automation .....</b>	<b>15</b>
Digging into Automated Patch Management Features .....	16
Integrating Tools with IT Infrastructure.....	18
Comparing the Pros and Cons of Automated Patching.....	18
Pros.....	19
Cons.....	19
Considering Tool Scalability and Customization .....	20

<b>CHAPTER 4:</b>	<b>Maintaining Security and Compliance</b> .....	23
	Meeting Compliance Standards .....	23
	Managed service providers .....	24
	Small and midsize businesses.....	24
	Highly regulated industries.....	25
	Auditing and Reporting Patch Records.....	27
	Responding to Vulnerabilities.....	27
	Applying Patch Management to a Corporate Security Strategy ...	28
<b>CHAPTER 5:</b>	<b>Optimizing Patch Management Configuration</b> .....	29
	Diagnosing and Resolving Common Issues .....	29
	Preparing Your Network, Systems, and Devices .....	30
	Step 1: Take an inventory of what you have.....	30
	Step 2: Analyze your network topology and segmentation.....	31
	Step 3: Develop a patch management policy .....	31
	Step 4: Deploy patch management tools.....	31
	Step 5: Create a patch testing environment.....	31
	Step 6: Establish baseline configurations .....	31
	Step 7: Implement backup and recovery procedures .....	32
	Step 8: Strengthen your security controls and monitoring mechanisms .....	32
	Step 9: User awareness and training.....	32
	Step 10: Pay attention to compliance and implement internal auditing.....	32
	Handling Failures and Rollbacks.....	33
	Scripting Custom Solutions .....	34
	Enhancing the Patch Process.....	34
<b>CHAPTER 6:</b>	<b>Presenting the NinjaOne Patch Management Solution</b> .....	35
	Gaining Insight into Overall Patch Status .....	35
	Saving Time with Automated Patching.....	36
	Creating Actionable Patch Reports .....	37
	Reaching All Endpoints .....	38
	Patching for Windows, macOS, and Linux.....	39
	Windows.....	40
	macOS .....	40
	Linux .....	41
	Applying Patches to Third-Party Applications.....	42
<b>CHAPTER 7:</b>	<b>Ten Features of Automated Patching</b> .....	43

# Introduction

Patch management is difficult at best, and at its worst, it is almost impossible to realize success. This is due, in part, to the continuous vigilance and patch cycle that we find ourselves in today. Automated patch management helps take the labor, time, and sting out of ongoing patch management, compliance, and reporting. This book covers general patch management issues and pain points and their resolutions using an automated, cloud-based, agent-driven solution.

## About This Book

*Patch Management For Dummies, NinjaOne Special Edition*, consists of seven chapters that explore the following:

- »» What patch management is and why it matters (Chapter 1)
- »» How to implement a patch management strategy (Chapter 2)
- »» Tools and automation your organization can use for patch management (Chapter 3)
- »» Security and compliance requirements (Chapter 4)
- »» How to optimize your patch management configuration (Chapter 5)
- »» The NinjaOne patch management solution (Chapter 6)
- »» Ten features of automated patching (Chapter 7)

Each chapter is written to stand on its own, so if you see a topic that piques your interest feel free to jump ahead to that chapter. You can read this book in any order that suits you.

## Foolish Assumptions

It's been said that most assumptions have outlived their usefulness, but I assume a few things nonetheless!

Mainly, I assume that you have some experience with the joys and non-joys of patch management, especially applying patches

manually. I also assume you've dealt with patching disasters, such as systems that won't return to an operational state, kernel panics, infinite reboots, and patches that won't uninstall. Patching is not the most exciting task that system administrators perform, but it is one of the most important. It is the criticality of patch management that makes us, as system administrators realize that automation is not only a necessity but also an inevitability.

If any of these assumptions describe you, then this is the book for you! If none of these assumptions describe you, keep reading anyway! It's a great book and after reading it, your knowledge of multi-cloud networking won't be cloudy!

## Icons Used in This Book

Throughout this book, I occasionally use special icons (I promise, no cutesy emojis) to call attention to important information. Here's what to expect:



REMEMBER

This icon points out important information you should commit to your nonvolatile memory, your gray matter, or your noggin.



TECHNICAL  
STUFF

This icon explains the jargon beneath the jargon and is the stuff legends — well, legendary nerds — are made of.



TIP

Tips are appreciated, but never expected — and I sure hope you'll appreciate these useful nuggets of information.



WARNING

These alerts point out the stuff your mother warned you about (well, probably not), but they do offer practical advice.

## Beyond the Book

There's only so much I can cover in this short book, so if you find yourself at the end of this book wondering, "Where can I learn more?" go to [www.ninjaone.com](http://www.ninjaone.com).

## IN THIS CHAPTER

- » Acknowledging the importance of patching
- » Introducing patch management
- » Separating patches by type
- » Discovering the patch management life cycle
- » Understanding and meeting the challenges

# Chapter **1**

# Introducing Patch Management

**Y**ou may assume that everyone understands the importance of patching. But the number of exploited vulnerabilities, compromised endpoints running outdated operating systems and applications, and systems hosting unsupported software tells you this isn't common knowledge.

This chapter covers the importance of patching and patch management, discusses the patching process, and explores its challenges.

## Realizing the Criticality of Patch Management

Patching, like backups and security, is vital for hardware, software, and operating systems, whether cloud-based or physical assets housed in a data center. Updating software and systems is the primary defense against software bugs and security vulnerabilities that lead to business interruption, loss of customer

confidence, and costly repair of damaged systems and applications. Regular patching reduces an organization's security risk by minimizing an application's or system's attack surface.



WARNING

Security researchers continually discover new vulnerabilities that developers must address. This ongoing cycle puts a strain on information technology (IT) budgets and personnel. A single missed or delayed patch may result in significant damage and business disruption.



REMEMBER

Everyone, from top organizational leaders to entry-level employees, must acknowledge the criticality of patching and allow IT teams the time to maintain systems and software properly.

## Defining Patch Management

*Patch management* is a centralized process for applying patches to IT assets. Patches improve security, enhance performance, and increase productivity. Patch management consists of downloading, installing, and checking vendor-supplied software fixes for vulnerabilities and bug fixes. System updates and upgrades are not the same as patches. A *patch* is a specific update type that addresses security concerns, flaws, and vulnerabilities. Patch management falls under the larger security umbrella of vulnerability management.

System administrators generally apply patches during outage windows, in the evenings, or, in the case of high-priority patches, during the workday. Applying patches may disrupt normal business flows and cause significant downtime for users and systems. Patch management suites aim to minimize downtime and negative impact on users and business operations.



REMEMBER

Even user-applied patches cause downtime for users. For example, a user receives a notice that patches for their workstation are now available. The patch process requires a reboot and time to install. User workflows are interrupted, and their productivity suffers.



TIP

A formal patch management process allows administrators to prioritize critical updates. The company benefits from patching with minimal disruption to employee productivity.

# PATCHING VERSUS UPGRADING

A patch is an upgrade that addresses a particular security flaw or vulnerability. Upgrades may provide new features, resolve functionality issues, or target security problems. Upgrades may also inadvertently include new vulnerabilities.

## Identifying Patch Types

When security personnel or IT administrators discuss patches, they generally refer to them generically as patches or updates. However, there are three types of patches:

- » **Security patches:** Security patches aren't released on a schedule but are provided by vendors to quickly target, address, and fix a particular vulnerability. Security patches are made available to customers free of charge for a limited product lifetime.
- » **Bug fix patches:** Bug fix patches mitigate problems in a software package. They're generally made available on a schedule or regularly as part of vendor maintenance as a benefit of an application subscription or purchase.
- » **Feature update patches:** Features update patches are provided to subscribers and released regularly, sometimes as optional but recommended minor or major version upgrades.

## Outlining the Patch Management Life Cycle

This section outlines the ten steps of the patch management life cycle with each step shown separately. A complete patch management life cycle includes these ten steps, whether combined or implemented separately, depending on an organization's workflows and patching activities.

## **Step 1: Identifying assets**

Inventorizing network assets is a crucial step in patch management because it identifies every asset that requires patching. Leaving one device or system unpatched provides an attack vector for malicious activities and compromise. This inventory process also informs the security team of network device vulnerabilities.

## **Step 2: Prioritizing patches**

Gathering a list of vulnerable devices allows the team to prioritize patching activities. Administrators categorize systems by risk level to determine how soon patches should be applied.

## **Step 3: Creating policies**

After your systems are categorized, you can create patching policies. These patching requirements, or criteria, determine what needs to be patched, when devices and applications should be patched, and under what conditions.

## **Step 4: Monitoring patch status**

In this step, administrators watch for new vulnerabilities and their patches from vendors. Organizations generally have a system for receiving notifications of upcoming patches and vulnerability updates from vendors instead of having to maintain manual vigilance.

## **Step 5: Testing patches**

Teams first install patches onto systems in a test environment to monitor unexpected issues before rolling out patches to demilitarized zones (DMZs), staging, development, user, and production environments. Testing allows administrators to exclude troublesome patches before pushing them to the wider network.

## **Step 6: Documenting changes**

Any changes to the patching process should be documented, and team members should be informed before patching activities. Any delay in critical patch deployment should be fully documented and include justifications and a modified plan.

## Step 7: Deploying patches

This is the formal patching step, where administrators deliver and install patches to systems according to the established patch policies outlined in Step 3. Any failures or issues arising in this step are assessed in the next step.

## Step 8: Auditing post-patch issues

Post-patch issues can arise, and environments should be monitored for performance problems and other anomalies. End users should be notified of any issues, changes, and upcoming solutions if necessary.

## Step 9: Reporting compliance

A monthly patch compliance report should be generated to allow executives and other departments to gain insight into your current IT infrastructure and how patching affects it.

## Step 10: Reviewing, updating, and repeating steps

The final stage of the patch management life cycle is to review, update, and repeat the preceding steps. This will keep information up to date and accurate, allowing an IT team to refine and optimize all patch management processes.

# Handling the Challenges

Five common patch challenges affect businesses that manage their infrastructure.

- » Time
- » IT inventory
- » Unsolved risks
- » Patch failures
- » Vulnerability management

According to a 2024 IT automation, security, and consolidation study by Omdia, an independent analyst and consultancy firm

that provides quantitative and qualitative insights into enterprise IT department challenges and priorities, only 37 percent of the 624 surveyed companies had implemented a strong patch management strategy ([https://go.ninjaone.com/leading\\_it\\_trends\\_2024\\_omdia\\_ninjaone](https://go.ninjaone.com/leading_it_trends_2024_omdia_ninjaone)). IT and cybersecurity professionals believe patching is too complex and time-consuming. To make patching more efficient, IT professionals aim to streamline and automate operations as much as possible. Gathering an accurate inventory of all systems, network hardware, and applications is difficult but necessary to manage vulnerabilities and keep systems updated.



WARNING

Emergency and priority patching can leave systems vulnerable by postponing medium- and low-priority patches. This practice weakens network security and increases business risks. Updating software is risky, and patch failures can cause numerous problems for an organization. According to Heimdal Security (<https://heimdalsecurity.com/blog/software-patching-statistics-practices-vulnerabilities>), “72 percent of managers are afraid to apply security patches right away because they could ‘break stuff.’” This patch reluctance often results in organizations adhering to an “N-1” patch policy. In other words, the latest patches are postponed in favor of patches from the previous wave of vulnerabilities.

Vulnerability management, including patch management, is a “catch-up” game because patches are issued after a breach or threat actors discover a vulnerability. Patches and upgrades sometimes introduce new vulnerabilities that must be patched.

Businesses and their IT and security teams have realized that meeting patch management challenges requires moving to an automated solution. The business costs and risks to critical systems and services are too high to leave to manual operations.

## IN THIS CHAPTER

- » Performing risk assessments
- » Working through the patch management process
- » Complying with regulations
- » Evaluating the process's effectiveness

# Chapter 2

# Implementing a Patch Management Strategy

This chapter examines patch management strategy, risks, policy development, compliance, and the complexities of the process in greater detail.

## Prioritizing and Assessing Risks

If your company and its assets have ever experienced a security breach or compromise, you understand how costly it is to repair the damage to systems, services, and reputation. Maintaining business operations is challenging, given the threat of ongoing attacks from hackers, malicious government-sponsored teams, insider threats, and corporate espionage. You also have to deal with power and hardware failures and human error.



REMEMBER

The risks to business operations are high, but knowledge is power, and prioritizing and assessing risks is the first step in maintaining continuity.

# Developing a Policy

A *patch management policy* is a guideline that ensures controlled, efficient, and secure patching. The guide contains steps and procedures that system administrators and security teams follow when patching vulnerabilities. It covers patching for a wide range of assets, which may include operating systems, software, applications, and network equipment.



TIP

For organizations with more than a handful of devices to manage, a patch management policy is a requirement for adherence to best vulnerability management practices. Even if a patch management policy isn't *necessary*, using one helps you map policy to regulatory requirements.

## Testing and Validating Patches

IT administrators test and validate patches to prove that applied patches and the patching process proceed as planned and don't introduce new errors, performance issues, or security flaws into an environment. Though the exact steps may differ from one organization to another, the following ten are generally accepted as standard procedure:

### 1. Create a test environment.

Establish a separate testing environment that mirrors the organization's production infrastructure as closely as possible. This environment should include representative systems, applications, and configurations to simulate real-world conditions.

### 2. Select test cases.

Identify test cases that represent common use cases and scenarios within the organization's information technology (IT) environment. To ensure comprehensive testing, test cases should cover a variety of operating systems, applications, and hardware configurations.

### 3. Test patch compatibility.

Install patches in the test environment and verify compatibility with existing software, applications, and configurations.

Test patches on different operating systems, versions, and hardware platforms to ensure broad compatibility across the organization's IT infrastructure.

**4. Assess patch impact.**

Evaluate patches' impact on system performance, stability, and functionality. Monitor key performance metrics such as central processing unit (CPU) usage, memory usage, disk input/output (I/O), and network traffic to identify any adverse effects caused by the patches.

**5. Test application functionality.**

Verify that patched applications and systems continue functioning as expected after patch installation. Conduct functional testing to ensure critical applications, services, and processes remain operational and perform their intended tasks without errors or disruptions.

**6. Conduct security testing.**

Assess the effectiveness of patches in addressing security vulnerabilities and mitigating potential security risks. Conduct vulnerability scanning, penetration testing, or security assessments to identify any remaining vulnerabilities or security weaknesses that may require further mitigation.

**7. Conduct user acceptance testing (UAT).**

Involve end users or stakeholders in the testing process to validate the impact of patches on their workflows, applications, and productivity. Solicit user feedback to identify any issues, concerns, or usability issues related to the patched systems or applications.

**8. Document test results.**

Document the results of testing activities, including observations, findings, and any issues encountered during the testing process. Keep detailed records of test cases, test procedures, test data, and test outcomes to facilitate analysis and validation.

**9. Validate the patch deployment process.**

Validate the patch deployment process to ensure patches are deployed correctly and according to established procedures. Verify that patch deployment tools, scripts, and automation processes function as intended and effectively deploy patches across the organization's IT infrastructure.

## 10. Review and iterate.

Review test results, identify areas for improvement, and iterate on the patching process as needed. Address any issues or deficiencies identified during testing, update test cases and procedures accordingly, and conduct additional testing to validate changes and improvements.



REMEMBER

This process is ongoing for teams that manage medium to large environments. Vendors regularly release patches, which must be tested, vetted, and confidently deployed in a production environment.

## Monitoring the Process

One of the pain points of patch management is monitoring devices and their patch statuses. Monitoring is painful because there are no native automated patch process tracking tools. Administrators must monitor progress manually by watching as patches are applied and checking logs after the fact should something go wrong. You can watch logs in real time on Unix, Linux, and similar operating systems. However, doing so requires singular focus, so patching multiple systems simultaneously is almost out of the question for an individual administrator.

Most monitoring is done by collecting and scraping logs after patching. Administrators scrape logs by searching for and collecting messages based on keywords, error codes, or statuses.

Long-term visibility is also critical to keeping systems updated and secured. Without constant monitoring capability, endpoints may drift out of compliance or succumb to an unpatched vulnerability, putting the entire network and data at risk.

## Reporting on Patch Compliance and Effectiveness

Do you have insight into how effective your patch management process is? Compliance reports include detailed text describing patch compliance levels, including the number of systems

patched, patch deployment success rates, and any outstanding vulnerabilities. These reports provide visibility into the organization's overall security posture. Gathering information on patch compliance involves scraping logs, directly observing, and comparing pre- and post-patching vulnerability scan results.

Administrators also include incident response procedures in patch compliance reporting to demonstrate how patching practices contribute to incident prevention and mitigation. They may also document any security incidents related to unpatched systems.

Finally, patch compliance reports include an executive summary highlighting key metrics, such as patch compliance rates, risk reduction due to patching, and any notable improvements in the security posture due to patch management efforts.

## IN THIS CHAPTER

- » Surveying automated patch management features
- » Assessing tools integration
- » Listing the pros and cons of automation
- » Addressing scalability and customization

# Chapter **3**

## Exploring Patch Management Tools and Automation

**S**ystem tools, scripts, scheduling programs, and applications can automate every step of the patching process. The central questions that arise in meetings about shifting to a more automated solution are

- » What happens when something fails?
- » How will we monitor the patching process?
- » Is our automation solution scalable?
- » How will an automation solution seamlessly integrate with our other tools and applications, ensuring compatibility and smooth operation?

This chapter covers patch management tools and automation, examining automated patch management features, integration with existing tools, the pros and cons of automated patching, and the flexibility and adaptability of automated solutions for scalability and customization.

# Digging into Automated Patch Management Features

Systems don't include native automated patch management features beyond using preconfigured software repositories and resolving software dependencies, so administrators must search for, test, vet, and implement applications to perform the required tasks. The features list varies from product to product, with some offering a "bare bones" toolset, while others provide applications and suites that include every possible tool, dashboard, and reporting function.



TIP

Several factors dictate which tools an organization selects to automate patching:

- » **Network size and complexity:** The larger and more complex an organization's network, the greater the need for sophisticated and full-featured tools.
- » **Administrator team size and skill set:** The size and collective skill set of the administrator team can affect the selection of the automation software suite.
- » **Budgetary constraints:** Departmental budgets dictate how much money is allocated toward capital outlay for new software.
- » **Compatibility with existing systems:** A significant issue is an application suite's flexibility and compatibility with existing infrastructure and team workflows.

Automated patching tools and suites offer several key features that help streamline the patch management process and ensure that systems are updated with the latest security updates. Some of the main features of automated patching tools include:

- » **Automated patch deployment:** Automated patching tools can automatically download and deploy patches to systems across the network, eliminating the need for manual intervention. This feature ensures timely vulnerability patching.
- » **Patch scheduling:** Users can schedule patch deployment during maintenance windows or off-peak hours to minimize

disruptions to business operations. Patch scheduling allows for efficient patch distribution and installation.

- » **Patch scanning and detection:** Automated patching tools can scan systems to detect missing patches and identify vulnerabilities that need to be addressed. This feature helps administrators prioritize patching based on criticality.
- » **Patch testing:** Some automated patching tools allow users to test patches in a controlled environment before deployment to production systems. Patch testing helps identify potential compatibility issues and ensures a smooth deployment process.
- » **Centralized patch management:** These tools provide a centralized console or dashboard for administrators to view and manage patching activities across all systems. Centralized patch management simplifies the oversight of patch compliance and reporting.
- » **Reporting and compliance tracking:** Automated patching tools generate reports on patch compliance levels, deployment status, and vulnerabilities addressed. Administrators can track the effectiveness of patching efforts and demonstrate compliance with security policies.
- » **Customization and configuration:** Users can customize patch deployment policies, define patch approval workflows, and configure settings based on organizational requirements. This flexibility allows for tailored patch management practices.
- » **Integration with established IT systems:** Automated patching tools may integrate with existing IT systems, such as configuration management tools, ticketing systems, and security information and event management (SIEM) solutions. Integration streamlines patch management workflows and enhances visibility.
- » **Patch rollback:** If a patch causes issues or conflicts with system functionality, automated patching tools may include a rollback feature to revert systems to a previous state. Patch rollback helps mitigate risks associated with faulty or failed patches.
- » **Comprehensive patch repository:** Automated patching tools maintain a comprehensive patch repository with the latest updates from software vendors. This ensures administrators have access to up-to-date patches for various applications and operating systems.



TIP

By leveraging these features of an automated patch management suite, organizations can more effectively manage the patching process, enhance their security posture, and reduce the risk of security breaches resulting from unpatched vulnerabilities.

## Integrating Tools with IT Infrastructure

Integrating new tools and applications with existing infrastructure and other tools is essential for adoption. A new application suite's flexibility in integrating into existing infrastructure and workflows is one of the deciding factors for teams considering an automated patching solution. No business executive wants an IT manager to approach them with a “rip and replace” or a costly additional infrastructure proposal to accommodate a new tool. Too many changes or costs that exceed budgets will cause executives to delay or cancel new tool purchases.



TIP

IT teams should check tool flexibility and integration by asking for case studies and customer references, which are valuable in decision-making.

Some changes — such as spinning up a new database server or web server or making other minor additions — are expected, but expensive or disruptive changes generally won't be acceptable. New tools may have hidden costs, such as administrator learning curves, workflow changes, and some unforeseen incompatibilities, but these costs are expected for any new tooling or application. Integration allows organizations to quickly deploy and start realizing the benefits of automation, reducing time to value and accelerating return on investment.

## Comparing the Pros and Cons of Automated Patching

Every technology solution has pros and cons, and automating patch management is no different. This section identifies the pros and cons of implementing an automated solution for your patch management activities.

## Pros

You can streamline your patch management process, reducing the time and effort required to deploy patches across a large network of systems. Automation eliminates many manual tasks, such as downloading patches and executing installation commands, leading to significant time savings for IT administrators.

Automated patching helps organizations avoid emerging threats by ensuring that systems are promptly updated with the latest security patches. By automating patch deployment and prioritizing critical patches, businesses can reduce the window of vulnerability and strengthen their overall security posture.

Businesses benefit from automation's consistency and reliability in patch deployment across the entire IT infrastructure. By following predefined deployment schedules, maintenance windows, and reboot policies, automated patching helps maintain system stability and reduce the risk of disruptions caused by inconsistent patching practices.

Automation is highly scalable and can accommodate the patching needs of organizations with diverse IT environments, including large-scale enterprise networks and distributed systems. Teams can handle patch deployment tasks across thousands of endpoints and devices, making automated patching tools well suited for organizations of all sizes.

Automated patch management solutions generate detailed reports on deployment status, compliance, and trends, enabling organizations to demonstrate adherence to patching policies and regulatory mandates.

Automated patching offers numerous efficiency, security, and scalability benefits. Organizations must carefully weigh the advantages and disadvantages to determine if automated patching is the right approach for their needs and IT environment. Planning, testing, dedicated vendor support, and ongoing monitoring are essential to maximize the benefits of automated patching while minimizing risks and potential drawbacks.

## Cons

Automated patching introduces the risk of automation errors, such as misconfigurations, compatibility issues, and unintended

consequences. If not adequately addressed, patch deployment scripts or automation workflow errors can lead to patch failures, system downtime, or security vulnerabilities.

Your automation scheme may overlook specific issues or dependencies that require manual intervention or validation. For example, patches that require specific prerequisites or compatibility checks may only be deployed correctly with manual oversight, leading to incomplete or ineffective patching.

Implementing automated patching solutions can be complex, requiring customization and configuration to fit the organization's specific requirements and IT environment. Configuring automated patching workflows, integrating with existing systems, and managing complex patching scenarios may require specialized expertise and resources.

Automation relies on patch management tools and software solutions, which may introduce dependencies, limitations, and vendor lock-in. Organizations must select and deploy patch management tools that align with their needs, budget, and technical requirements.

Unfortunately, automated solutions are not “set it and forget it” because they can disrupt business operations if not carefully managed. Patch deployments scheduled during peak hours or without proper testing and validation may cause system downtime, application failures, or performance issues, leading to productivity losses and user dissatisfaction.

## Considering Tool Scalability and Customization

IT teams must consider their scalability and customization options when selecting new tools. Any new tools should be able to adjust to a business's size and complexity without multiple instance purchases. (This assertion may have exceptions depending on how the vendor licenses its software, but this section focuses on the broad spectrum of cases and not exceptions.) Tools should be flexible enough to expand and scale as business requirements change.



*Scalability* refers to the ability of a system, application, or infrastructure to handle increasing workloads, user demands, or data volumes without experiencing a significant decrease in performance, reliability, or efficiency. IT resources must adapt to changing business requirements, accommodate growth, and effectively meet the needs of users and applications.

Creating a tool that’s “one size fits all” is challenging, but it should be possible to add modules, plug-ins, or additional infrastructure to support an ever-expanding network and larger data-sets. Having these customizable options is an essential factor in tool choice. The worst scenario is when an IT team makes a tool purchase and realizes the tool has some upper limit to the number of users, devices, or systems it can address, or there are no options for customizing the software to handle special cases such as field offices, roaming users, wireless devices, or devices to which an agent can’t be installed.

## IN THIS CHAPTER

- » Following industry-specific compliance standards
- » Reporting patching procedures and results
- » Developing a preemptive approach to vulnerability management
- » Integrating patch management into corporate security

# Chapter 4

# Maintaining Security and Compliance

**R**egulatory compliance requires organizations to adhere to relevant state, federal, and international laws, security frameworks, and industry mandates. Compliance consumes considerable time, labor, consulting fees, licensing fees, audits, technology, and fines for the most regulated industries, such as healthcare, manufacturing, insurance, banking, and finance.

This chapter covers how meeting those standards affects some key industries, such as managed service providers (MSPs), small and midsize businesses (SMBs), and the highly regulated health-care, manufacturing, and financial industries.

## Meeting Compliance Standards

Organizations are subject to many evolving laws and standards. Meeting compliance standards helps ensure consumer safety and confidence, system and data security, and privacy. Additionally, these standards safeguard the organization against legal repercussions and foster a culture of responsible governance.

## Managed service providers

An MSP may seek System and Organization Controls (SOC) compliance to satisfy customer demands and ensure the company can deliver its services and service-level agreements (SLAs). SOC compliance requires that the company submit to an audit of its operations. An SOC audit involves a third-party auditor who validates the service provider's controls and systems.



TECHNICAL  
STUFF

SOC compliance isn't typically mandatory in specific industries, like Payment Card Industry Data Security Standard (PCI DSS) compliance, which is used to process payment card data. In general, companies need a SOC audit when their customers request it.

There are three levels of SOC compliance, and some providers hold more than one SOC compliance certification:

- » **SOC 1** focuses on controlling access to clients' financial information.
- » **SOC 2** is a more general assessment of the service provider's controls for various trust services criteria (TSCs), including security, availability, privacy, processing integrity, and confidentiality. The security TSC is mandatory, while the others are optional. SOC 2 compliance certification is essential for MSPs.
- » **SOC 3** compliance affects the general public rather than businesses.

Type I audits determine compliance at a single point in time; Type II audits determine compliance over time. For MSPs, the SOC 2 Type II audit is the most common. SOC 2 certification gives the MSP's customers a higher level of trust that the security measures employed by the MSP are sufficient to protect customer data and services.

## Small and midsize businesses

Compliance varies for SMBs depending on how many employees they have, their annual revenue, and the industry they're a part of. Companies must generally adhere to all local, state, and federal regulations that govern their operations. Compliance may mean paying employees correctly, filing taxes on time, providing benefits, following appropriate hiring practices, and keeping accurate records.

# Highly regulated industries

Some industries require extensive compliance reporting because they deal with consumer privacy information, financial data, health and safety, or direct healthcare delivery. The following are examples of industries requiring reporting compliance activities and their results:

» **Healthcare:** Patch compliance reporting requirements for the healthcare industry are essential for ensuring the security and integrity of sensitive medical data and systems. Regulatory standards and best practices govern these requirements to safeguard patient information and maintain healthcare systems' confidentiality, integrity, and availability.

Healthcare organizations are subject to multiple regulatory agencies and standards, such as the Health Insurance Portability and Accountability Act (HIPAA) in the United States, which mandates implementing security measures to protect patient information. These regulations may not specifically mention patch management, but they typically require organizations to implement safeguards to protect against security vulnerabilities, including timely patching of software and systems.

Patch compliance reporting requirements for the healthcare industry emphasize the importance of proactive risk management, timely patch deployment, and adherence to regulatory standards to protect patient information and maintain the security of healthcare systems and infrastructure.

» **Insurance:** Insurance companies must comply with various regulations to protect customer data, ensure privacy, and safeguard against cyberthreats. Regulatory standards such as HIPAA, the Gramm–Leach–Bliley Act (GLBA), and state-specific data protection laws establish cybersecurity and data privacy guidelines, often including patch management and compliance reporting requirements.

» **Finance and banking:** Financial and banking institutions must follow stringent regulatory requirements to safeguard financial transactions, prevent fraud, and protect customer information. Regulatory bodies such as the Federal Financial Institutions Examination Council (FFIEC) in the United States and the European Banking Authority (EBA) in the European

Union establish guidelines and standards for cybersecurity and risk management, which often include requirements related to patch management and compliance reporting.

» **Manufacturing:** Regulatory requirements for manufacturers are generally related to cybersecurity, data protection, and product safety. Depending on the industry and geographic location, regulations such as the General Data Protection Regulation (GDPR), the California Consumer Privacy Act (CCPA), and industry-specific standards like the International Organization for Standardization (ISO) 27001 may establish guidelines for patch management and compliance reporting.

Affected facilities must generate patch compliance reports to demonstrate adherence to patch management policies and regulatory requirements. These reports typically include details such as the status of patch deployment across the organization, patch coverage for critical systems and applications, compliance with patch deployment timelines, and any outstanding patches or vulnerabilities.

Patch compliance reporting may involve conducting regular vulnerability scans and assessments to identify security vulnerabilities and assess the effectiveness of patch management efforts. These scans help organizations prioritize patches based on the severity of vulnerabilities and ensure comprehensive coverage of systems and applications.

Records include audit trails and documentation of patch management activities, including patch deployment schedules, testing results, mitigation strategies for critical vulnerabilities, and any deviations from established patch management policies. Additionally, these records help demonstrate due diligence in addressing security vulnerabilities and mitigating risks.

Organizations relying on third-party vendors for software and services may have contractual obligations to ensure patch compliance and security. These requirements may include regular audits of vendor patch management practices, verification of patch compliance for vendor-supplied software, and collaboration with vendors to address security vulnerabilities and apply patches in a timely manner.

# Auditing and Reporting Patch Records

Auditing the patch management process is key to verifying its efficiency and compliance with set standards and protocols. The initial step involves scrutinizing the documentation related to patch management, which includes strategies, methods, and change records, to gain insights into the entity's patching approach. Next, assessing the practices for deploying patches is crucial to guarantee that updates are applied promptly and extensively across all systems. A thorough review of the logs and reports related to patching can highlight any discrepancies or omissions within the process. Conducting vulnerability assessments and penetration testing is also vital for detecting any potential threats.



TIP

The best patch management tools automatically generate patch reports demonstrating compliance across all devices in the information technology (IT) environment by monitoring each device's patch status. These reports provide a comprehensive overview of patch status across all endpoints within an organization. IT administrators demonstrate an organization's commitment to maintaining a secure IT infrastructure during audits and internal compliance reviews by generating detailed reports listing applied patches, pending updates, and noncompliant devices.

Patch management reports offer insights into patching status across servers, workstations, and mobile devices. By analyzing these reports, IT administrators can identify systems running outdated software or missing critical patches, enabling them to prioritize patch deployment based on vulnerability severity and reduce the overall attack surface.

## Responding to Vulnerabilities

There are two different methods of responding to vulnerabilities. The first is a manual response, which consists of security team members keeping up to date on vulnerabilities from various sources, such as vendor sites, government notifications, and security organizations. This method requires time and vigilance, perhaps from multiple team members searching documentation, email messages, and posted notifications. The probability of missing a critical vulnerability is high.

Alternatively, a team may automatically be notified when a device, operation system (OS), or third-party application vulnerability is detected using automation. Policies and custom scripts can automatically prioritize mitigation efforts based on critical vulnerability scores. The team may leverage automation driven by policies and powerful scripting capabilities to perform mitigation actions at scale efficiently. The likelihood of missing a critical vulnerability is low.

## Applying Patch Management to a Corporate Security Strategy

A good patch management policy that complies with industry regulations should be part of a corporate security strategy because patch management is more broadly included in vulnerability management. Vulnerability management covers system hardware, applications, data, operating systems, network appliances, printers, disaster recovery, mobile phones, tablets, and any device with network or internet access. It applies to the entire corporate security profile.



REMEMBER

Integrating patch management into a corporate security strategy is essential for protecting against vulnerabilities, reducing the risk of cyberattacks, and ensuring the security and integrity of an organization's IT infrastructure and its users.

## IN THIS CHAPTER

- » Preparing systems for patching
- » Connecting every endpoint
- » Implementing a plan of action
- » Developing custom solutions
- » Improving the process through automation

# Chapter 5

# Optimizing Patch Management Configuration

**F**or more than a few systems, an IT administration team should formalize the patch management process, including optimizing their systems for patching. Performing manual patching is time-consuming and tedious, so preparing your network, systems, and devices is essential to successful patch deployment, compliance, and reporting.

This chapter covers preparing your network and endpoints by finding system problems, collecting information, anticipating patch failures, creating custom solutions, and reviewing and improving the process.

## Diagnosing and Resolving Common Issues

One of the main problems facing IT teams that want to implement a formal patch management process is device connectivity. In rare situations, systems must be air-gapped or otherwise

isolated. Those systems must also be patched, but they're the exceptions to the larger patch management rollout plan. Isolated systems and devices should have a separate patch management plan and process.

During the inventory and information-gathering phase of implementing a patch management process, teams must locate, repair, replace, and reconnect all devices to the network.

Another common problem with devices is age. End-of-life devices are no longer supported with patches and are vulnerable to attack. If such devices are not replaceable, they must be segregated from the rest of the network either physically or logically.

Another problem facing IT teams is network complexity. This issue is difficult, if not impossible, to resolve. Networks grow and evolve, but each segment should be assessed independently to simplify unraveling a complex network.

Handling remote devices is a pain point that adds complexity to a corporate patch management strategy. Devices in field offices and employees with devices rarely touching a local network must also participate in patch management.

## Preparing Your Network, Systems, and Devices

Use the following ten steps as a guide to prepare your systems and devices and formalize the patch management process.

### Step 1: Take an inventory of what you have

Conduct a comprehensive inventory of all network systems, devices, and software applications within the organization's information technology (IT) infrastructure. Maintain an up-to-date inventory database that includes each asset's hardware specifications, software versions, configurations, and patch statuses.

## **Step 2: Analyze your network topology and segmentation**

Analyze the organization's network topology and segmentation to identify critical network segments, subnets, and zones that require patch management. Implement network segmentation to isolate critical systems and applications from less secure or non-essential network components.

## **Step 3: Develop a patch management policy**

Develop a patch management policy that outlines the processes, procedures, and responsibilities for identifying, testing, deploying, and monitoring patches across the organization's network systems and devices. Define patch deployment schedules, maintenance windows, and reboot policies based on business requirements and operational needs.

## **Step 4: Deploy patch management tools**

Select and deploy patch management tools and software solutions compatible with the organization's network infrastructure, systems, and applications. Evaluate patch management solutions based on features such as automated patch deployment, centralized management, reporting capabilities, and integration with existing IT systems.

## **Step 5: Create a patch testing environment**

Create a dedicated patch testing environment or sandbox to simulate patch deployment scenarios and validate patch compatibility, functionality, and stability before deployment to production environments. Use test systems and virtualized environments to conduct patch testing without impacting production systems.

## **Step 6: Establish baseline configurations**

Establish baseline configurations for network systems and devices to ensure consistency and standardization across the

organization's IT infrastructure. Document baseline configurations for operating systems, applications, and network devices, and use configuration management tools to enforce and maintain baseline settings.

## **Step 7: Implement backup and recovery procedures**

Implement backup and recovery procedures to safeguard critical data, configurations, and system settings before patch deployment. Perform regular backups of network systems and devices to ensure data integrity and facilitate recovery in case of patching errors or system failures.

## **Step 8: Strengthen your security controls and monitoring mechanisms**

Strengthen security controls and monitoring mechanisms to detect and mitigate security risks associated with patch management. Implement intrusion detection systems (IDSs), network monitoring tools, and security information and event management (SIEM) solutions to monitor network traffic, detect anomalies, and respond to security incidents.

## **Step 9: User awareness and training**

Educate network administrators, IT staff, and end users about the importance of patch management and cybersecurity best practices. Provide training and awareness programs to help users recognize security threats, report vulnerabilities, and follow patch deployment procedures.

## **Step 10: Pay attention to compliance and implement internal auditing**

Ensure compliance with regulatory requirements, industry standards, and internal policies related to patch management. Implement auditing mechanisms to track patch deployment activities, monitor patch compliance, and generate reports for audit and compliance purposes.

# Handling Failures and Rollbacks

Failures are inevitable, but they don't have to be disastrous if you create a plan to monitor, resolve, and prevent their frequency. In the case of unresolved failures, patches must be rolled back to restabilize the system or device in question.

The progress of patch deployment should be monitored to identify any errors, failures, or issues that may occur during the patching process and track patch deployment status, system logs, and error messages in real time.

Team members should assess the impact of the patch failure on system functionality, performance, and security; determine the severity of the issue; and evaluate the potential risks associated with leaving the system in its current state without remediation. If possible, isolate affected systems from the network to prevent further disruptions or security risks while troubleshooting and resolving patch failures. Finding failure root causes may involve verifying patch compatibility, resolving configuration conflicts, updating device drivers, or applying manual fixes to resolve errors or issues.



REMEMBER

If patch failures can't be resolved promptly, administrators must begin a rollback procedure to revert the system to its previous state before the patch deployment. You may have to use backup and recovery procedures to restore system settings, configurations, and data to the pre-patch state. You should then test and validate the rollback procedure and verify that the system has been successfully reverted to its previous state without any lingering issues or side effects. Patch failures should be documented, along with the rollback procedure and resolution steps, for future reference and audit purposes.

Finally, the team should implement preventive measures to reduce the likelihood of patch failures in the future. This may include better research, improving patch testing procedures, implementing change management processes, enhancing backup and recovery capabilities, and implementing monitoring and alerting mechanisms to proactively detect and respond to patching issues.

# Scripting Custom Solutions

IT departments often create internal patch repositories so when their systems need patches, they can only connect to and retrieve them from those custom sources. Custom scripting solutions for patching involve creating scripts or automation workflows to streamline the patch management process, customize patch deployment procedures, and address specific requirements or challenges within an organization's IT environment. So-called "homegrown" automated systems are challenging to manage, complex to maintain, and often not secure. However, custom solutions persist because of the need to automate parts of the patching process. Some IT departments combine multiple tools and applications to help streamline, automate, and simplify patch management.

## Enhancing the Patch Process

Because patch management is time-consuming, complex, and a continuous cycle, systems administrators attempt to enhance the patching process by centralizing patch management operations. They use centralized management tools to monitor patch deployment progress, track patch compliance, and generate reports on patching status. As stated earlier, these solutions are likely a mixture of homegrown scripts and commercial applications that provide limited but otherwise missing functionality.

## IN THIS CHAPTER

- » Managing every endpoint
- » Realizing automation's return on investment
- » Reporting patch status and compliance
- » Patching third-party applications

# Chapter 6

## Presenting the NinjaOne Patch Management Solution

This chapter brings together the information in this book to feature NinjaOne's specific offerings in automated patch management, generating patching and compliance reports, connecting every endpoint, and handling patch management for third-party applications.

### Gaining Insight into Overall Patch Status

NinjaOne's intuitive patching dashboard provides clear insight into your organization's overall patch status and enables your admins to make faster, more informed decisions to improve patch compliance. Quickly and easily view operating system (OS) patch status for all Windows, macOS, and Linux endpoints to make faster, more informed decisions.

Because NinjaOne's endpoint management solution is cloud based and agent deployed, your internet-connected endpoints

are always visible and manageable regardless of location — even behind firewalls.

## Saving Time with Automated Patching

Manual patching is tedious and time-consuming. Administrators must connect to each endpoint individually, download and install patches, reboot the system, wait for the system to recover successfully, check for more patches, and repeat this process until there are no more patches to install. Some patching policies require the administrator to “check out” the patched system by opening common applications such as an email app, internet browser, word processing applications, spreadsheet programs, and other software that must be fully functional.

Switching to an automated system results in time savings of as much as 90 percent over manual patching. Automated patch management allows users to set up patching schedules and ensure that updates are pushed uniformly to all endpoints without significant human intervention.

### AUTOMATED VERSUS AUTOMATIC PATCHING

*Automated* patching automatically identifies, downloads, tests, deploys, and manages software patches across an organization’s IT infrastructure without manual intervention.

*Automatic* patching applies patches as they become available through a scheduled process, such as a cron job that automatically downloads and installs patches and any associated dependencies without manual intervention.

Automated patching differs from automatic patching, but the two terms can be confusing. Automatic patching bypasses testing by allowing a scheduled task to update the system, often setting the system to reboot automatically if required. This process can leave systems unusable if something goes wrong with the update.

Conversely, automated patching applies all necessary patches that system administrators have tested and vetted.

Automatic patching can have extremely negative consequences because the system applies patches as they become available through the vendor without any testing. This sounds like a good idea unless a patch is applied to a system that causes conflicts with an application, OS, service, or hardware.

**Remember:** Patches should always pass through a “vetting” process before being applied to production or other critical systems.

## Creating Actionable Patch Reports

Actionable patch reports provide valuable insights into the organization’s patching status, including patch deployment progress, compliance levels, and vulnerabilities addressed. By analyzing these reports, IT administrators and security teams can make informed decisions about patch prioritization, resource allocation, and risk mitigation strategies. Here are some specific advantages of creating actionable reports:

- » Simplifying patch management through centralized reporting
- » Automating patch deployment and ensuring efficient updates without manual intervention
- » Delivering comprehensive reports on patch status and compliance and offering actionable insights
- » Providing real-time monitoring of patching progress for proactive remediation
- » Mitigating security risks by promptly addressing vulnerabilities across the network
- » Facilitating compliance with industry regulations and internal policies
- » Boosting operational efficiency by streamlining reporting and workflows and reducing manual tasks

By “actionable,” I mean that taking actions based on report findings will help you in the following ways:

- » Regularly schedule automated scans to identify vulnerabilities proactively.
- » Prioritize patch deployment based on severity levels and criticality to minimize risk.
- » Utilize customized reports to track patching progress and compliance.
- » Implement role-based access controls to restrict permissions and ensure data security.
- » Conduct regular audits to verify patching efficacy and address any gaps.
- » Stay informed about emerging threats and industry best practices to adapt your patch management strategy accordingly.

Overall, actionable patching reports empower organizations to optimize their patch management processes, enhance security, and demonstrate compliance with regulatory requirements, ultimately reducing the risk of security incidents and ensuring the integrity and availability of their IT infrastructure.



REMEMBER

Patching is the single most critical aspect of a device hardening strategy. According to Ponemon, effective patching can prevent almost 60 percent of breaches.

## Reaching All Endpoints

*Endpoint visibility* is the ability to view, monitor, and manage all the endpoints in your IT environment. You accomplish this goal using endpoint management software. Tracking your endpoints also enables you to take necessary action, such as securing them or updating them for better performance.

NinjaOne’s agent-based approach enables complete management of any internet-connected device so you can quickly and securely support remote and hybrid employees.

Endpoint visibility allows organizations to manage their devices effectively. Seeing every endpoint is extremely helpful from a security perspective. You know all the components you're responsible for and what you must protect. Without that knowledge, you're taking a guesswork approach to endpoint security. You can't know whether all the devices connected to the organization are accounted for and secured. If you can't see it, you can't secure it.

Endpoint visibility can give you immediate knowledge when something goes wrong, even if the end user doesn't detect anything. When a problem is detected, a best practice is to remediate it as soon as possible. This helps prevent an issue from escalating and causing further damage to an endpoint.

Monitoring and managing all your devices will also inform you whether your endpoints are in good or bad health. Endpoint visibility is crucial to the health and security of your organization's technology. It also allows you to support your end users better. Because of its importance, you want to ensure it's set up correctly and provides you with the critical endpoint data you need.



TIP

One of the best ways to achieve endpoint visibility is by using endpoint management software. This type of software is specifically designed to monitor and manage all endpoints connected to your business's network. It also allows you to access your endpoints, apply endpoint security, install applications on devices, change configurations, provide support, and more.

## Patching for Windows, macOS, and Linux

NinjaOne's patch management solution covers all Windows, macOS, and Linux systems via locally installed agents and the convenience of cloud connectivity. Endpoints will stay monitored and updated anywhere there's an internet connection.



TECHNICAL  
STUFF

Agent-based solutions have existed for more than 20 years. The best large-scale example of agent-based monitoring is anti-malware software that uses an agent to connect to a central service to support malware signature updates. Agents are an efficient method of protecting, updating, and maintaining systems.

# Windows

Microsoft provides Windows Update for operating system update management, but the main difference between Windows Update and a patch management tool is the intended use and capabilities. Windows Update is a Windows operating system component designed to keep Windows computers updated with the latest patches and security fixes directly from Microsoft. It's suitable for individual users or small networks.

In contrast, Windows patch management software is designed for larger organizations and provides more robust features. These tools allow IT administrators to manage and deploy updates across multiple systems and applications, not just Windows. They offer scheduling, reporting, and compliance management features essential for Windows server patch management and maintaining the security and efficiency of an organization's IT infrastructure. Windows patch management tools provide a centralized platform for administrators to control the entire patching process, ensuring that all systems, including servers, desktops, and mobile devices, are up to date and secure.



TIP

NinjaOne allows you to take complete control over your Windows patch management settings. You can set automatic patch approval settings for each patch type and criticality. You also get full control over scanning and updating schedules, reboot options, and more.

## macOS

Organizations can utilize macOS patch management solutions designed to manage updates across multiple endpoints from a single console to centralize the patching of macOS-based devices. These solutions automate identifying available patches, deploying updates, and ensuring compliance with patching policies. Additionally, mobile device management (MDM) platforms often include features for managing macOS devices, including patching.

Another option is to leverage third-party patch management tools tailored explicitly for macOS environments. These tools streamline the patching process, allowing administrators to schedule updates, track patch status, and ensure that all macOS endpoints are promptly patched to maintain security and compliance.



NinjaOne allows you to control your macOS patch management settings. You can set automatic patch approval settings and have full control over scanning and updating schedules, reboot options, and more. Save time and improve patch compliance by remotely patching your macOS devices automatically with NinjaOne. Automate patch scanning and set update schedules for all your macOS computers to minimize time spent on patch management.

## Linux

Even the most skilled IT departments and managed service providers (MSPs) often need help with patch management problems. You may only be able to solve some of these Linux patching challenges at one time, but being aware of their existence is the first step in creating a safer, more efficient patch management process. Here are some Linux patching challenges and their resolutions:

- » **Workflow disruptions:** When patching endpoints affecting large groups of people, such as Linux servers, IT teams must schedule patch rollouts during off-peak hours. By scheduling this way, organizations can avoid disrupting workflows with normal patching processes, such as reboots.
- » **Imperfect patches:** Unfortunately, patches aren't perfect. Even the patches that undergo rigorous sandbox testing sometimes create bugs that must be fixed. One way to ensure that a patch functions as it should is to install it on a small group of your Linux devices rather than your entire IT infrastructure. If the small group has no negative effects from the update after a certain period, then installing the patch to the rest of the Linux endpoints is usually safe.
- » **Volume of patches:** Patching every Linux device on a network takes time, especially if an organization doesn't use automation, and it's complex work. Small businesses may not have a problem, but large organizations and enterprises often struggle with the enormous volume of patches that must be deployed.
- » **Manual mistakes:** Human errors and manual mistakes happen; other than automation, there is no way to prevent them entirely. The consequences of unpatched software are often severe, and all a cybercriminal needs to succeed is one forgotten and unpatched Linux server or endpoint.



TIP

NinjaOne provides clear insight into the Linux distributions currently supporting your IT estate. Make faster, more informed decisions to protect your users and maintain a strong security posture. You can also patch any application available via Advanced Package Tool (APT), Yellowdog Updater Modified (YUM), or Dandified YUM (DNF) in the repository configured on your endpoint.

## Applying Patches to Third-Party Applications

Patch hundreds of common business applications across operating systems automatically to remove known vulnerabilities. NinjaOne's third-party patching engine cuts software deployment time and minimizes vulnerabilities in business software for Linux, Windows, and macOS systems. NinjaOne allows you to automatically install and patch more than 135 of the most common business applications without end-user intervention.

Within NinjaOne, the Third-Party Patch Management feature provides a centralized platform to ensure the currency of all your software applications, including popular third-party ones like Adobe, Chrome, Java, and more. By automating the patching process, your organization maintains protection against potential vulnerabilities, effectively minimizing the risk of cyber threats.



TIP

Learn more about NinjaOne Patch Management at [www.ninjaone.com/patch-management](http://www.ninjaone.com/patch-management). Schedule a live tour at [www.ninjaone.com/contactusform](http://www.ninjaone.com/contactusform). Or start your free trial of the NinjaOne platform at [www.ninjaone.com/freetrialform](http://www.ninjaone.com/freetrialform).

# Chapter 7

## Ten Features of Automated Patching

Automation has many advantages over performing tasks manually, and automated patching includes all those advantages. Here are ten features that come with deploying NinjaOne's automated patch management solution:

- » **Patching dashboard:** Identify known vulnerabilities and deploy patches at scale across servers, workstations, and laptops to minimize your attack surface. Leverage NinjaOne's intuitive patching dashboard to quickly view patch status for all Windows, macOS, and Linux endpoints through a dashboard showing the patch status of all your devices under management. See which devices have pending patches, failed patches, and installed patches in an easy-to-read dashboard.
- » **Operating system (OS) patching:** NinjaOne's patching solution supports various operating systems and applications, making it versatile for diverse IT environments. Automating patch identification, approval, and installation across Windows, macOS, and Linux servers and workstations can minimize known vulnerabilities.
- » **Patch automation:** Automated security patch management streamlines the process of deploying patches, reducing the

time and effort required to keep systems up to date. You can patch endpoints 90 percent faster with zero-touch patch identification, approval, and deployment. Automate every step of the patching process so your technicians can focus on support and strategic projects.

- »» **Cloud-first, agent-based:** Get better patch compliance for remote endpoints via cloud-based patching — no company network, domain, or virtual private network (VPN) required. NinjaOne is cloud-based and agent-deployed, so you can patch any endpoint using an internet connection.
- »» **Remediation tools:** You can efficiently remediate patching and other device issues by using NinjaOne's built-in remote terminal, registry editor, and remote access tools.
- »» **Preemptive patch approval:** Preemptively approve patches to prevent zero days, and automatically block problem patches to avoid service outages.
- »» **Reboot management:** Improve patch compliance, drive technician efficiency, and provide a better user experience with automated reboots.
- »» **Vulnerability data:** Get visibility into endpoint security with per-patch data on known vulnerabilities, including knowledgebase articles, common vulnerabilities and exposures (CVE) bulletins, and Common Vulnerability Scoring System (CVSS) scores.
- »» **Alerts and notifications:** Continuous monitoring for available patches ensures you're always aware of necessary updates. Instantly notify technicians via email, SMS, Slack, and other channels of pending or failed patches for faster remediation.
- »» **Patch reporting:** Quickly and easily report on patch compliance status, failed patch deployments, and known endpoint vulnerabilities at the click of a button. Get comprehensive insights into the status of all patches deployed across your network, including successful installations, pending updates, and any issues you encounter. Streamline your reporting process with automated generation of patch management reports, saving time and effort. Gain granular visibility into patch deployment activities, allowing you to track updates on individual devices or device groups.

# Manage, patch, and support all your endpoints

Patch management is challenging at best, but at worst, it's an ongoing labor-intensive struggle that, if not done correctly, can result in security breaches, system compromises, and data theft. Automating patch management can cure the common ills of the patch process by providing full visibility of all network assets and their current patch statuses and compliance levels.

## Inside...

- Learn the benefits of automated patch management
- Minimize the possibility of human error
- Maximize your security posture
- Satisfy your security compliance requirements
- Maintain a watchful eye on all your endpoints
- Track patch deployment in real time
- Deploy patch policies across all devices

## ninjaOne®

**Kenneth Hess** is a Linux and Windows System Administrator, open-source software advocate, technology journalist, filmmaker, and author.

Go to **Dummies.com™**  
for videos, step-by-step photos,  
how-to articles, or to shop!

ISBN: 978-1-394-27006-4

Not For Resale



for  
**dummies®**  
A Wiley Brand

# **WILEY END USER LICENSE AGREEMENT**

Go to [www.wiley.com/go/eula](http://www.wiley.com/go/eula) to access Wiley's ebook EULA.