



PATCH MY PC

Patch Perfect:

A Practical Guide
to IT Resilience

Patch Management Best Practices:

2. Introduction: A Step-by-Step Guide
3. Why Patch Management Matters
4. Updating is Critical: Statistics and Facts
5. Common Techniques
6. 8 Steps for Common Best Practices — Steps 1-2
7. Best Practices — Step 3
8. Best Practices — Steps 4-5
9. Best Practices — Steps 6-8
10. Additional Best Practices for Effective Patch Management
11. Conclusion: Building a Resilient Patch Management Strategy

A Step-by-Step Guide

Patch management is essential for a secure, stable, and compliant IT environment, but managing it effectively is often easier said than done. IT teams face overwhelming workloads, complex patching schedules, and constant pressure to prevent security breaches and downtime. Without a structured approach, patching can feel like an endless cycle of catching up, exposing organizations to security threats, compliance issues, and costly disruptions.

This guide offers more than just a checklist. It's a proven roadmap that simplifies patch management across workstations, servers, and third-party applications. Backed by insights from industry best practices and field-tested methods, each step is designed to streamline patching, reduce workload, and help your IT team stay ahead of potential vulnerabilities.

Dive in, and in just eight steps, you'll uncover practical strategies that can turn patching from a chore into a game-changing asset for your IT operations.

Why patching?



Vulnerabilities



Exploits by
Hackers



Malware /
ransomware



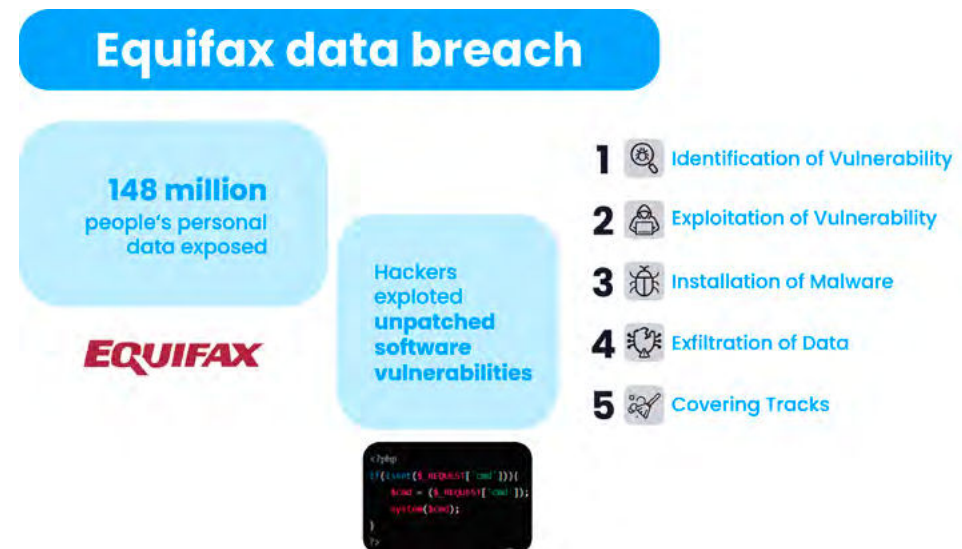
Data breach

Why Patch Management Matters

Before we dive into the practical steps, let's look at why patch management is crucial for any IT environment. Cyber threats are on the rise, and many data breaches occur due to unpatched vulnerabilities. Just look at cases like the Equifax breach in 2017, where a missed patch exposed sensitive information for millions of users. These real-world examples underscore why timely patching isn't just ideal; it's essential.

In the next section, we'll get into how automation can help make patching more manageable and consistent.

- **Security:** Patches close security gaps discovered by vendors and developers. Without timely patching, these vulnerabilities are open doors for attackers to exploit, leading to data breaches and potential financial and reputational damage.
- **Stability and Performance:** Many patches fix bugs or enhance features, improving the overall performance of applications. Consistent patching helps prevent system crashes and maintains smooth operations, directly impacting productivity.
- **Compliance:** Industries like healthcare, finance, and government mandate regular updates to comply with regulatory standards (e.g., HIPAA, GDPR). Failing to patch can lead to compliance violations and hefty fines.
- **Business Continuity:** Unpatched vulnerabilities can cause unexpected outages, leading to operational downtime and potential loss of revenue. A solid patch management strategy minimizes these risks.



Updating is critical

57%

of data breaches could have been prevented by being patched on time

11 seconds

companies are hit by ransomware attempt

ISO 27001

and Cyber Essentials certification patching is required

\$4.4 Million

average cost of a data breach

Now that we understand the importance of patch management, it's time to explore the core techniques that can streamline this process. Patching can be tedious and time-consuming, so knowing your automation, testing environments, and scheduling options is critical. Ready to find out how automation can free up time for your team while keeping systems secure?

— *Let's dive in.*

Automate Patch Management

The average time to patch a critical vulnerability is around 16 days, and many organizations face delays due to manual patching. Automation is essential for maintaining timely patching and reducing the workload on IT teams. An automated patching system can handle updates without manual oversight, ensuring regular compliance.

Benefits of Automated Patch Management

- **Consistency:** Automation enforces routine patching cycles, reducing human error and ensuring critical patches are not missed.
- **Efficiency:** Automated tools can check for new patches, test, and install them while ensuring minimal disruption to user productivity.
- **Compliance:** Automating patch schedules improves patch compliance, which is crucial for regulatory standards and audit readiness.

Testing Patches in a Controlled Environment

Testing patches before deployment is critical to prevent compatibility issues and maintain system stability. Certain patches may conflict with applications or cause other unforeseen issues, so testing updates in a controlled environment is always recommended.

- **Create a Test Environment:** A test environment should mirror your production network and include all operating systems in use. This helps ensure patches work as expected across different devices and setups.
- **Focus on Critical Updates First:** Prioritize testing for critical patches that address high-severity vulnerabilities, especially on business-critical and internet-facing systems. Non-critical updates can follow according to a regular schedule.

Testing in a controlled setting helps prevent disruptions in production and minimizes the risk of patch-related issues affecting end users. Let's explore how to keep patching efficient with a twice-weekly schedule.

Schedule Patching Twice a Week

Patch frequency is crucial for maintaining security. While monthly or quarterly cycles are common, a twice-weekly patching schedule (e.g., Mondays and Thursdays) offers greater protection, especially as vendors frequently release new patches for popular applications. Next, we'll look at a structured approach to patch management that will take this strategy even further.

Benefits of Twice-Weekly Patching

- **Improved Security:** Frequent patching reduces the window of exposure for vulnerabilities.
- **Consistent Protection:** Twice-weekly schedules keep systems up-to-date with the latest security updates, especially for applications like browsers that are frequently targeted by attackers.
- **Reduced Impact on User Productivity:** By automating patching twice a week, IT teams ensure a regular rhythm without overwhelming systems or causing bottlenecks.

Effective patch management is more than just scheduling updates and automating tasks; it's about building a solid foundation that keeps your IT environment secure, compliant, and running smoothly. With a proactive strategy, you can go beyond just "keeping up" with patches to getting ahead of vulnerabilities and minimizing disruptions. While scheduling and automation are key components, the following eight steps take patch management further by providing a strategic approach covering everything from prioritizing critical updates to ensuring compliance and minimizing user impact. Let's dive in and see how these best practices can transform your patch management process.

Step 1: Assess Your Patch Management Goals and Environment

Before implementing patching configurations, **it's crucial to evaluate your organization's needs and set clear patching goals:**

- **Identify Critical Assets:** Identify essential devices, servers, and applications that are critical to daily operations. This allows you to prioritize updates for assets that, if compromised, would have a high business impact.
- **Define Security and Compliance Standards:** Review your industry's regulatory and security requirements to establish patching policies that align with both security and compliance needs.
- **Set Patch Prioritization Criteria:** Not every patch is equally critical. Establish prioritization criteria based on threat level and asset importance, so high-severity patches on critical systems are addressed promptly.

Step 2: Design Your Patch Management Strategy

A well-defined strategy tailored to different device types and applications is essential for effective patch management.

1. Separate Processes for Workstations, Servers, and Applications:

- **Workstations:** Workstations are central to user productivity, so schedule updates in phases to minimize disruptions and address potential issues early.
- **Servers:** Schedule server updates during off-peak hours or maintenance windows to ensure availability for critical applications.
- **Microsoft Office and Third-Party Applications:** Non-Microsoft applications like Adobe Acrobat or Google Chrome are common targets for attackers. Include these in your patching strategy, ensuring both Office and third-party applications are secure.

2. Choose Appropriate Tools: Tools like Windows Update for Business (WUfB), WSUS, and SCCM automate patching for Windows OS and select applications, but they don't cover everything. This is where third-party tools like Patch My PC come into play, taking on the heavy lifting for non-Microsoft applications and making it easier for IT teams to keep systems fully patched without the extra hassle.

3. Develop a Consistent Update Cycle: With a twice-weekly schedule, ensure critical updates are applied promptly, reducing the risk of missed patches and minimizing user disruptions.

Step 3: Implement Update Rings with Windows Update for Business (WUfB) in Intune

Just like we use [update rings in Patch My PC Cloud](#), using rings for WUfB is a smart move. Update rings let you roll out patches in stages across workstations, reducing risk by identifying any issues in smaller test groups before deploying broadly.

1. Create Update Rings:

- **Pilot Ring:** This ring includes a small group of non-critical devices that receive updates first, enabling early detection of issues.
- **Early Adopters:** This group includes more devices, allowing for additional feedback on update stability.
- **Broad Deployment:** The final ring includes the remaining devices, where updates should be stable and tested by this stage.

Feature Deferral	Quality Deferral	Feature	Quality	Servicing channel
0	0	Running	Running	Retail Channel
7	3	Running	Running	Retail Channel
14	7	Running	Running	Retail Channel

2. **Incorporate Reboot Scheduling:** Once updates are staged by ring, it's essential to time reboots effectively. Weekly reboots can be scheduled over weekends, prioritizing Sunday (46%) and Saturday (27%) for minimal disruption. For daily updates, evening hours (57%) ensure devices restart without interrupting peak work hours. Automated reboots help maintain compliance and complete updates without relying on user action.

3. **Configure Deadlines and Deferrals:** Deadlines specify a final date by which updates must be installed, even if users delay them. Deferrals delay updates to allow IT to observe effects in each ring and ensure stability before broader deployment.
4. **Encourage Active Hours Configuration:** Allow users to set active hours, reducing unexpected reboots during peak productivity. Default active hours can be set across the organization, but user adjustments personalize the experience.
5. **Manage Power Settings:** Configuring power settings prevents devices from sleeping or shutting down during update installations, ensuring patches apply successfully. Refer to Step 6 for detailed guidance on configuring power settings, such as preventing sleep and hibernation modes during updates
6. **Proactive Notifications:** Notify users about scheduled updates and potential reboots well in advance. Intune allows configuring notifications that inform users when updates are scheduled, giving them time to save work and prepare.

Step 4: Use Windows Server Update Services (WSUS) for Granular Control

WSUS provides centralized control for approving and distributing patches, making it ideal for on-premises or hybrid environments. Even with [Microsoft's plans to deprecate WSUS](#), it remains a valuable tool for environments that

rely on centralized patch management, especially for managing internal Windows updates.

- **Centralized Patch Management:** WSUS allows you to view, approve, and prioritize patches, controlling critical systems and delaying less essential changes.
- **Patch Testing and Approval:** Test patches in a controlled environment before deployment to minimize risks.
- **Patch Approval Policies:** Set automated policies for high-priority patches while manually reviewing others for greater control.

If you're looking to get started or need tips on optimizing WSUS, check out this comprehensive [WSUS configuration and clean-up guide](#) to make the most of your setup.

Step 5: Establish a Patch Testing and Approval Process

Testing patches before deployment is essential to catch potential compatibility issues and maintain system stability.

1. Set Up Testing Environments:

- **Workstations:** Use a small group of pilot devices to test updates, ensuring minimal impact if issues arise.
- **Servers:** A separate environment that mimics production can reveal potential impacts on system load or software compatibility before full deployment, ensuring minimal impact if issues arise.

2. **Approval Workflows:** High-impact patches may require additional review before deployment. Define workflows for testing, approving, and expediting critical patches while allowing time for careful review of less urgent updates.

3. **Use Phased Rollouts in SCCM:** SCCM's phased rollout feature allows gradual deployment to servers, reducing the risk of widespread issues by applying updates progressively.

We take a similar approach at Patch My PC for your third-party updates, testing each update in our labs across multiple OS environments and running each installer through VirusTotal's 70+ antivirus engines for added security. You can find more details on our testing process in our [FAQ](#).

Step 6: Power Management for Workstations

Effective power management ensures workstations stay online for scheduled updates, especially in remote and hybrid setups.

- **Prevent Sleep and Hibernation:** Configure power settings in Intune to keep devices online during scheduled updates and ensure they receive patches.
- **Run Updates During Off-Peak Hours:** Scheduling updates outside peak hours minimizes disruptions, which is particularly valuable for remote environments.
- **Monitor Battery Health with Advanced Insights:** In remote scenarios, where devices may rely on battery power, it's essential to ensure that battery performance remains strong. With Patch My PC's Advanced Insights, you can monitor battery health to confirm devices are performing optimally, so you're not left vulnerable by devices that can't stay powered through critical updates. Learn more about monitoring battery health in the [Advanced Insights documentation](#).

Step 7: Organize Server Patch Management with SCCM

Servers are critical assets, requiring careful handling to avoid unplanned downtime.

1. **Group Servers by Role and Criticality:** Categorize servers by function, ensuring critical production servers receive updates more frequently than test or backup servers.

2. **Define Maintenance Windows:** Maintenance windows in SCCM specify when patches can be applied, ensuring updates don't disrupt core business hours.
3. **Plan Rollback Options:** For critical patches, SCCM's rollback feature enables IT to revert updates in case of issues, ensuring quick recovery.

Step 8: Automate Compliance Monitoring and Reporting

Automated monitoring ensures that all systems are consistently updated, simplifying compliance tracking and reporting.

- **Use Dashboards for Real-Time Tracking:** Dashboards in tools like Patch My PC and SCCM display patch status, making it easy to spot non-compliant devices.
- **Enable Non-Compliance Alerts:** Set automated alerts for devices that miss critical patches, enabling IT to address vulnerabilities quickly.
- **Generate Reports for Audits:** Document patch history, testing outcomes, and compliance records to streamline audit processes and demonstrate security efforts.

Experience intuitive reporting with rich, high-quality visuals that can be interpreted easily with [Advanced Insights](#).

Additional Best Practices for Effective Patch Management

Incorporating these best practices can improve the overall efficiency and reliability of patch management efforts:

- **Prioritize Critical Updates:** To minimize security threats, address high-risk vulnerabilities first. Non-essential updates can be scheduled for later.
- **Automate Where Possible:** Automation reduces manual effort and enhances consistency in patching. Use automated compliance tracking and notifications and update schedules wherever feasible.
- **Establish a Routine:** Consistent weekly (preferable, two patch deployments scheduled a week) or monthly patch schedules help set expectations, making updates predictable for end users.
- **Leverage Update Rings, WUfB, and WSUS for Phased Rollouts:** Phased rollouts and test environments allow IT to manage risk and ensure patches are stable before broad deployment.
- **Communicate with Users:** Inform users of scheduled patches, especially if reboots are required. Clear communication about the benefits of patching can reduce resistance and encourage compliance.
- **Maintain Detailed Documentation:** Documenting your patch management processes, test steps, and configurations is essential for audits, compliance, and continuity planning.

Conclusion:

Building a Resilient Patch Management Strategy

Effective patch management is crucial to protecting against vulnerabilities, ensuring compliance, and maintaining system stability. By following these best practices and implementing a structured approach, organizations can safeguard their IT environments and minimize the impact of patching on daily operations.

For a third-party patch management solution that automates, schedules, and alleviates many IT team issues, consider Patch My PC. A [free demo](#) will reveal affordable pricing and endless opportunities.