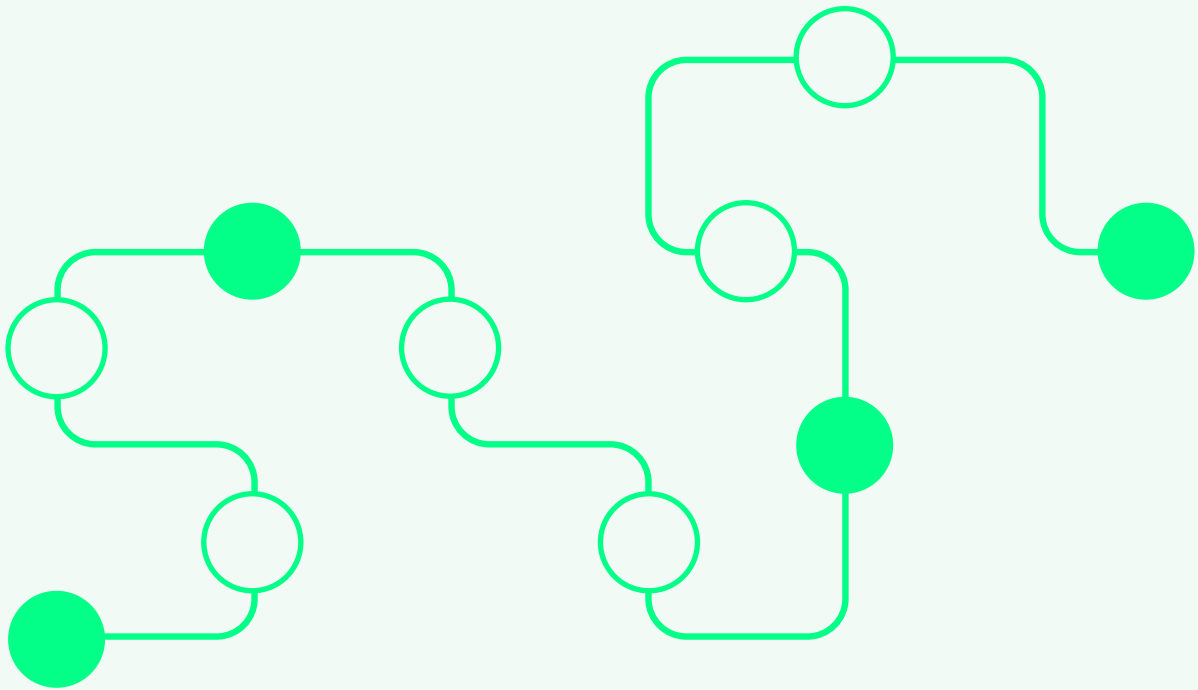


# ninjaOne®



# 10 Steps to Autonomous Patching

---

See where you are today and what it takes to reach full autonomy with NinjaOne.

Manual patching is a time sink and leaves you exposed. Basic automation still asks IT to babysit approvals, reboots, and updates. Use this checklist to see how far you've progressed toward autonomous patching and where NinjaOne can help you go further.

---

## Checklist: Where are you today?

### Step 1: Find the gaps

- I've documented manual patching tasks that drain IT time.
  - I can identify endpoints most exposed to patching delays.
  - **Next action:** Centralize visibility. Start tracking time spent, risk exposure, and failure rates to build the case for automation.
- 

### Step 2: See every vulnerability in real time

- I've integrated scanner data (Qualys, Tenable, Rapid7) into my patching workflow.
  - I can automatically map CVEs to affected endpoints.
  - **Next action:** Allow CVSS scores to inform patch prioritization.
- 

### Step 3: Deploy with control

- I categorize devices by risk level and business criticality.
- Bad patches are contained before they reach critical systems.
- **Next action:** Auto-approve patches and remove if issues are detected.



#### MILESTONE REACHED

Once the boxes are checked, you've mapped your patching pain points and started building control into deployments. Next up: use intelligence to reduce risk and start reclaiming IT time.

#### Step 4: Let AI protect you from bad patches



- I use AI to block or pause risky updates before they disrupt users.
- I receive proactive alerts for high CVSS patches.

→ **Next action:** Combine AI stability scoring with automated approvals for a faster, safer cycle.

#### Step 5: Stop wasting time on approvals



- Critical patches are auto approved.
- Known-problem updates are automatically blocked.

→ **Next action:** Layer in category-based rules (security vs. feature updates) to streamline approvals.

#### Step 6: Reboot without disruption



- Reboot policies adapt to user activity.
- Patch-related tickets have dropped significantly.

→ **Next action:** Use branded notifications and escalation rules to maintain security without breaking workflows.

#### Step 7: Prove compliance automatically



- I can monitor patch compliance in real time.
- Compliance reports are generated automatically.

→ **Next action:** Standardize reporting policies to enforce SLAs and meet audit requirements with no extra effort.



#### MILESTONE REACHED

Once the boxes are checked, you've automated the basics and are running smoothly. Next up: strengthen resilience with self-healing workflows and tighter security alignment.

### Step 8: Fix issues before they reach IT



- I can initiate rollback of failed or unstable updates.
- Scripts and workflows are reusable across environments.

→ **Next action:** Expand self-healing workflows across device types to eliminate recurring patch issues.

### Step 9: Make Security and IT one team



- Patch status feeds directly into my security and compliance workflows.
- Patch SLAs are enforced automatically by severity.

→ **Next action:** Align patch cadence with vulnerability management priorities for true IT + SecOps collaboration.

### Step 10: Get smarter every patch cycle



- My patch automation policies improve every cycle.
- Each round of patching is faster, safer, and less manual.

→ **Next action:** Build a feedback loop with NinjaOne analytics. Refine approval thresholds and policies to get smarter every cycle.



#### MILESTONE REACHED

Once the boxes are checked, you've arrived at autonomy. Patching is now intelligent, self-correcting, and fully integrated with security operations.

## Score your progress

### Count the number of boxes you checked:

**0-6 boxes:** You're still using manual patching or basic patching automation. NinjaOne accelerates your progress with unified policies, deployments, and CVE mapping.

**7-14 boxes:** You've automated some tasks, but risk remains. NinjaOne's Patch Intelligence AI, smart approvals, and compliance-ready reporting close the gap.

**15-20 boxes:** Nearly autonomous. NinjaOne eliminates the last-mile risks of patching by blocking unstable updates, auto-healing failures, and giving IT + SecOps shared visibility so even at scale, patching stays secure and disruption free.

NinjaOne turns patching into a hands-off, self-improving workflow that cuts wasted hours, reduces patch-related tickets, and gives IT back the bandwidth to focus on strategic work.

[Learn more](#)

[Contact us](#)