



PatchMon Release Notes

Version-by-version PatchMon release notes — features, fixes, breaking changes, and migration pointers.

LAST UPDATED

25 April 2026

READ ONLINE

<https://patchmon.net/docs/patchmon-release-notes>

CHAPTERS

6

SOURCE

github.com/PatchMon/PatchMon

Contents

1. [Table of Contents](#)
2. [Version 2.0.0](#)
3. [Version 1.4.2](#)
4. [Version 1.4.1](#)
5. [Version 1.4.0](#)
6. [Version 1.3.7](#)

Each section below documents what changed in a PatchMon release. Versions are listed newest first. The same source files are also served by the application in the admin UI under Release Notes.

Table of Contents

[Version 2.0.0](#)

[Version 1.4.2](#)

[Version 1.4.1](#)

[Version 1.4.0](#)

[Version 1.3.7](#)

Version 2.0.0

Architectural changes

Go

Backend rewritten from the ground up in Go, replacing Node.js and Next.js. The stack is more scalable, uses less RAM, and is significantly more performant.

[sqlc](https://sqlc.dev/) (<https://sqlc.dev/>) is used for type-safe SQL against PostgreSQL (compile-time checked queries instead of ad-hoc ORM access patterns).

[golang-migrate](https://github.com/golang-migrate/migrate) (<https://github.com/golang-migrate/migrate>) is used for database migrations (replacing Prisma ORM).

Structured logging with the standard library `log/slog` for cleaner, machine-parseable logs in production.

Background jobs and automation

Background work is handled by [asynq](https://github.com/hibiken/asynq) (<https://github.com/hibiken/asynq>) on Redis instead of BullMQ. PatchMon no longer ships the embedded **Bull Board** stack; queue visibility and triggers live in the existing **Automation** UI, which reduces attack surface, image size, and operational complexity.

Docker

Docker is the officially supported deployment method going forward; bare-metal installs are discontinued. A migration document describes the upgrade path.

Hardened base images are used. They ship with near-zero CVEs and a smaller footprint.

No separate frontend container: static React build artifacts are embedded in the Go binary. The container runs that binary (by default on port 3000) with `chi` (<https://github.com/go-chi/chi>): `/api/*` is handled by the server, so nginx inside PatchMon is no longer required. You still use nginx or another reverse proxy in front for TLS termination and public access as usual. A Guacamole (guacd) sidecar is included for Windows RDP. It is separate for now; RDP/VNC for Windows is an area we intend to improve.

API documentation

OpenAPI 3 spec is served at `/api/v1/openapi.json`, with **Swagger UI** under `/api/v1/api-docs` (authenticated) for exploring integration endpoints.

New features

Linux patching: Deploy updates per host or in bulk, on demand or on a schedule. **Policies** support host/group assignments and exclusions; runs support **approval**, **stop**, **retry validation**, and **live log streaming** over WebSocket.

Microsoft Windows agent (beta) and **FreeBSD** agent support.

Windows Updates (beta Windows agent): server APIs for update results, reboot state, superseded cleanup, and approved-guid sync, aligned with the new Windows agent.

Advanced monitoring & alerting: richer alert lifecycle (including assignment and bulk actions), optional **advanced alert configuration** for tuning and cleanup where your edition includes it.

Notifications: first-class **destinations** (SMTP, webhooks, ntfy), **routes**, **delivery log**, and **scheduled reports** so operational signals leave PatchMon reliably.

Environment variables in the GUI: many settings that were previously only in process environment can be **viewed and edited from the Settings UI** (per-key updates, with sensible validation), so you change less by hand in compose or shell env for day-to-day tuning.

OIDC / SSO: configure OpenID Connect from the same Settings area, including **import from environment** when you are migrating from a file-based or container env setup.

Other improvements

Compliance / OpenSCAP: SSG and CIS benchmarking content is **bundled in the server binary** at build time. Agents no longer pull scanning content from GitHub; everyone shares one versioned source of truth and less outbound traffic from agents.

SSO: improved sign-in flows and **Entra ID** integration compared to 1.4.x OIDC edge cases (e.g. redirect loops with auto SSO).

Dashboard: additional cards and data surfaces; dashboard layout preferences carry forward in the new UI.

Host integration config: **apply pending config** from the server so integration changes are applied to agents in a controlled, observable way.

Settings reliability: server URL and related configuration are reimplemented on the Go stack with database-backed resolution, addressing classes of “settings did not persist” issues from the Node era.

Reverse proxy awareness: continued correct use of forwarded headers for HTTPS/WSS behind proxies (without the Bull Board-specific HTTP quirks from 1.4.x).

Optional admin pprof: when enabled, CPU/memory profiling endpoints are available to administrators for performance investigation.

Packaging and editions

Features are grouped into **capability modules** (e.g. patching policies, advanced alerts, custom branding, Docker inventory, compliance depth, AI assist, remote access). Core workflows stay simple; larger deployments can enable more surface area where their **subscription or license** allows. See in-app **Context** / billing documentation for your tenant.

Known issues

Remote Desktop (RDP): there is a known bug with the RDP connection flow in this release. A fix is planned for the next release.

Migrations

This covers migration for Docker, Proxmox community scripts, and legacy `setup.sh` installs:

[Migrating from 1.4.2 to 2.0.0 \(https://docs.patchmon.net/books/patchmon-application-documentation/page/migrating-from-142-to-200\)](https://docs.patchmon.net/books/patchmon-application-documentation/page/migrating-from-142-to-200)

Version 1.4.2

□ PatchMon 1.4.2

Dashboard and UI

Editable dashboard: Dashboard widgets can be edited and re-arranged; a default layout is provided and editing is the default experience.

Bull Board missing over HTTP: The queue monitoring UI (Bull Board) did not appear when the app was served over HTTP (e.g. dev or internal HTTP). It now shows correctly for both HTTP and HTTPS.

Ultrawide (21:9) layouts: Dashboard layout is adjusted for 21:9 and similar ultrawide screens so content uses space better.

📄 Compliance

“Transaction already closed” errors: Compliance operations could fail with “Transaction already closed: A query cannot be executed on an expired transaction”. The underlying transaction/upsert handling is fixed so these errors no longer occur under normal use.

Stuck compliance scans: Scans that ran for 3+ hours could leave jobs in a “running” state. Automatic cleanup now stops and cleans up these long-running scans.

Cancel running scans: You can cancel a compliance scan that is in progress instead of waiting for it to finish or timeout.

Compliance dashboard and tables: Compliance dashboard rework: new dashboard card, clearer tables for scan results, and scanner status stored per agent/host. Table display and behaviour are improved.

Debian compliance scans: Fixes for Debian-related compliance scans so they run and report correctly.

Per-host scanner toggles: OpenSCAP and Docker Bench can be enabled/disabled per host. OpenSCAP defaults to on when compliance is on; Docker Bench defaults to off. Existing data is preserved via migration.

Log safety in compliance routes: Host IDs are sanitised before being written to logs so user-controlled input cannot inject fake log lines (e.g. via newlines).

📄 HTTPS and reverse proxy

WebSocket shown as insecure (ws) when using HTTPS: When PatchMon was behind a reverse proxy (e.g. nginx, Traefik) with HTTPS, the UI could still show the agent connection as insecure (`ws` instead of `wss`). This is fixed by correctly using `X-Forwarded-Proto` (including `https` and `wss`) and the header name used by some proxies (`http_x_forwarded_proto`), so the secure state matches how users actually connect.

📄 OIDC and authentication

OIDC login/logout loop: With “auto redirect to OIDC” enabled, some users hit a redirect loop between login and logout. That flow is fixed so OIDC-only setups work as intended.

Auto-redirect to OIDC: When `OIDC_ENABLED=true` and `OIDC_DISABLE_LOCAL_AUTH=true` , the app now automatically redirects to the OIDC provider instead of showing the local login page.

⚙️ Settings and URL config

Settings and URL not saving: Server URL and related settings (protocol, host, port) could fail to save or be lost after restart. The backend now uses the database as the source of

truth for the server URL after initial creation, so URL and environment-related settings persist correctly and are retrieved when loading the settings page.

Agent and hosts

Agent download from GitHub: Fixes for downloading agents from GitHub so installs/updates complete reliably.

NanoPi / no disks: On devices like NanoPi with no disks (or when no disks are detected), the UI could show “null” or errors. Disk handling and display are fixed for “no disks” cases, and related lint issues are addressed.

Docker endpoint agent update: The non-fatal warning during agent update in the Docker endpoint was removed to reduce noisy logs.

Agent log sanitisation: OpenSCAP agent logs sanitise output so user-controlled or command output cannot inject newlines into log streams.

Integrations

Checkmk: You can export hosts from the Integrations page for use with Checkmk.

Discord OAuth2: Discord login and account linking are supported. The Discord OAuth callback was also updated for CodeQL and security (e.g. no raw OAuth parameters in logs, proper validation).

Security and dependencies

NPM vulnerabilities: Dependency bumps and fixes to address known NPM vulnerabilities.

License: License is clearly stated as AGPL v3 in the repo.

Code quality and secrets: Code scanning and CodeQL are enabled.

Other

Fonts: Fonts are self-hosted where applicable for faster load and fewer external requests via DNS.

Biome: Linting/tooling uses a pinned Biome version for consistent formatting and checks.

Thank you

I appreciate the whole community for helping with PRs and help testing areas of PatchMon <3

Version 1.4.1

▣ PatchMon 1.4.1

A maintenance release with OIDC improvements, FreeBSD agent support, installer fixes, and various bug fixes and improvements.

▣ OIDC Improvements and Hot Fixes

- OIDC authentication fixes and stability improvements

- Hot fixes for edge cases in SSO flows

📱 FreeBSD Agent Support

- Native FreeBSD agent support** — run the PatchMon agent on FreeBSD hosts

- Initial FreeBSD support via community contribution

📦 Native Installer Upgrade Fixes

- Fixes for native installer upgrade paths

- Improved reliability when upgrading existing installations

▣ Host Table Views Not Saving → Bug Fix

- Fixed an issue where host table view preferences (columns, sort order, filters) were not being saved

- Table view state now persists correctly across sessions

▣ Agent Memory Leaks and Improvements

- Addressed memory leaks in the agent

- General agent stability and resource usage improvements

🔑 Better API Integration Scoping

- Improved scoping for Integration API credentials and access

- Tighter integration between API keys and their permitted scope

▣ Acknowledgements

- @RuTHlessBEat200** — for agent and OIDC fixes

- @mminkus** — for FreeBSD initial PR

- The rest of the community for their support and help on Discord and GitHub

Version 1.4.0

▣ PatchMon 1.4.0

A major release with security compliance scanning, OIDC SSO, an alerting engine, web SSH terminal, and AI-assisted terminal support.

♥ Security Compliance Scanning

OpenSCAP CIS Benchmark scanning directly from the agent (Level 1 / Level 2)

Docker Bench for Security when Docker integration is enabled

Compliance dashboard with fleet-wide scores, pass/fail breakdowns, and scan history

Optional auto-remediation of failed rules during scans

▣ OIDC Single Sign-On

OpenID Connect authentication with Authentik, Keycloak, Okta, or any OIDC provider

Automatic user provisioning on first OIDC login

Group-based role mapping from your identity provider to PatchMon roles

Option to disable local auth and enforce SSO-only login

▣ Alerting & Reporting

New Reporting page with filtering by severity, type, status, and assignment

Host Down alerts real time view of host uptime

Alert types including server update, agent update, and host down

Per-alert-type configuration for default severity, auto-assignment, escalation, and retention

🖥️ Web SSH Terminal

Browser-based SSH to any host from the PatchMon UI

Direct and proxy modes (proxy mode routes through the agent, no SSH port exposure needed)

▣ AI Terminal Assistant

AI chat panel inside the SSH terminal for command suggestions and troubleshooting

Multiple providers supported: OpenRouter, Anthropic, OpenAI, Google Gemini

Context-aware using your recent terminal output

📄 UI Improvements

Toast notifications replacing disruptive `alert()` popups

Error boundary with crash recovery and a copyable error report

"Waiting for Connection" screen with real-time status when onboarding a new host
Swagger / OpenAPI docs served at `/api-docs` on the server

□ Other

Superuser management permission (`can_manage_superuser`) for finer-grained RBAC

More Scoped API stats and details on hosts with added flags such as `?include=stats` or `?updates_only=true`

Plus Much Much More

Version 1.3.7

□ **ALERT : Auto-update of Agent issue**

Versions <1.3.6 have an issue where the service does not restart after auto-update. OpenRC systems are unaffected and work correctly. This means you will unfortunately have to use `systemctl start patchmon-agent` on your systems to load up 1.3.7 agent when it auto-updates shortly.

Very sorry for this, future versions are fixed - I built this release notes notification feature specifically to notify you of this.

□ **New Features & Improvements :**

Mobile UI: Mobile user interface improvements are mostly complete, providing a better experience on mobile devices.

Systemctl Helper Script: In future versions (1.3.7+), a systemctl helper script will be available to assist with auto-update service restarts.

Staggered Agent Intervals: Agents now report at staggered times to prevent overwhelming the PatchMon server. If the agent report interval is set to 60 minutes, different hosts will report at different times. This is in the `config.yml` as `"report_offset: nxyz"`

Reboot Detection Information: Reboot detection information is now stored in the database. When the "Reboot Required" flag is displayed, hovering over it will show the specific reason why a reboot is needed (Reboot feature still needs work and it will be much better in 1.3.8)

JSON Report Output: The `patchmon-agent report --json` command now outputs the complete report payload to the console in JSON format instead of sending it to the PatchMon server. This is very useful for integrating PatchMon agent data with other tools and for

diagnostic purposes.

Persistent Docker Toggle: Docker integration toggle state is now persisted in the database, eliminating in-memory configuration issues. No more losing Docker settings on container restarts (thanks to the community for initiating this feature).

Config.yml Synchronization: The agent now writes and compares the `config.yml` file with the server configuration upon startup, ensuring better synchronization of settings between the agent and server.

Network Information Page: Enhanced network information page to display IPv6 addresses and support multiple network interfaces, providing more comprehensive network details.

Auto-Update Logic Fix: Fixed an issue where agents would auto-update even when per-host auto-update was disabled. The logic now properly honors both server-wide auto-update settings and per-host auto-update settings.

Prisma Version Fix: Fixed Prisma version issues affecting Kubernetes deployments by statically setting the Prisma version in package.json files.

Hiding Github Version: Added a toggle in Server Version settings to disable showing the github release notes on the login screen

The open source Linux patch management platform

PatchMon gives sysadmins one dashboard for patching across Linux, FreeBSD, and Windows fleets. Run it as a managed SaaS on PatchMon Cloud (per-host billing, 14-day trial, no fleet minimum) or self-host the AGPLv3 Community Edition on your own infrastructure.

[Start a trial: patchmon.net/pricing](https://patchmon.net/pricing)

