Users, Roles & SSO

- Setting Up Microsoft Azure Entra ID (SSO)
- Two Factor Authentication
- Users, Roles and RBAC

Setting Up Microsoft Azure Entra ID (SSO)

This is a step-by-step guide for configuring **Microsoft Azure Entra ID** (formerly Azure Active Directory) as the Single Sign-On provider for PatchMon using the **Settings UI**. No `.env` editing is required.

What You'll End Up With

- Users sign in to PatchMon with their Microsoft work account.
- PatchMon accounts are created automatically on first login.
- PatchMon roles (Super Admin / Admin / Host Manager / User / Readonly) are driven by Entra ID **security groups**.
- Optionally, local username/password login is disabled, so SSO is the only way in.

Everything is configured through **Settings → OIDC / SSO** in the PatchMon web interface.

Before You Begin

You'll need:

Item	Notes
A running PatchMon instance	Reachable at a fixed URL, e.g. `https://patchmon.example.com`
HTTPS on your PatchMon URL	Entra ID **will not** accept plain `http://` redirect URIs (except `http://localhost`)
An existing admin account in PatchMon	So you can sign in and open Settings. If you don't have one, complete the normal setup wizard first
Access to the Microsoft Entra admin center	`https://entra.microsoft.com`. You need **Application Administrator** or **Global Administrator** role on the tenant

Open two browser tabs side-by-side:

- **Tab 1:** PatchMon → sign in as admin → **Settings → OIDC / SSO**

- **Tab 2:** https://entra.microsoft.com

You will collect **six values** in Tab 2 and paste them into Tab 1:

1. Tenant ID
2. Application (client) ID
3. Client secret (the **Value**, not the Secret ID)
4. Admin group Object ID
5. User group Object ID
6. (Optional) any additional role group Object IDs

Part A: Configure Entra ID (Tab 2)

Step 1: Get the Callback URL from PatchMon First

Before you start in Entra, grab the callback URL PatchMon will use. You'll paste it into Entra.

1. In Tab 1, go to **Settings → OIDC / SSO**.
2. Scroll down to the **OAuth2 Configuration** section.
3. Look at the **Callback URL** field. It will say something like:

   ```
   https://patchmon.example.com/api/v1/auth/oidc/callback
   ```

4. Copy it. You'll need this in the next step.

> **Note:** This field is read-only and is derived from the PatchMon server URL setting. If it looks wrong (e.g. `http://localhost:3000` when you're running in production), fix your **Server URL** in Settings → General first.

Step 2: Register an Application in Entra ID

1. In Tab 2, open **Identity → Applications → App registrations**.
2. Click **+ New registration**.
3. Fill in the form:
   - **Name:** `PatchMon` (purely cosmetic, shown on the consent screen)
   - **Supported account types:** choose **Accounts in this organizational directory only (Single tenant)** for most deployments. Only pick multi-tenant if you explicitly

want users from other Entra tenants to sign in.
- **Redirect URI:**
  - Platform: **Web**
  - URL: paste the callback URL you copied in Step 1

4. Click **Register**.

You'll land on the app's **Overview** page. Copy these two values into a scratch note:

- **Application (client) ID**
- **Directory (tenant) ID**

Step 3: Create a Client Secret

1. In the left menu, open **Certificates & secrets**.
2. Under **Client secrets**, click **+ New client secret**.
3. Description: `PatchMon`. Expiry: pick a duration that fits your rotation policy (up to 24 months).
4. Click **Add**.
5. **Copy the `Value` column immediately.** This is the only time Entra will show it.

> **Do not** copy the `Secret ID`. That is a metadata GUID, not the secret. You want the `Value` column.

Save this value in your scratch note as **Client Secret**.

Step 4: Configure Token Claims (Add Groups)

PatchMon maps Entra ID groups to PatchMon roles, so Entra must include group information in the ID token.

1. In the left menu, open **Token configuration**.
2. Click **+ Add groups claim**.
3. Tick **Security groups**. Leave the other checkboxes unticked unless you specifically use Directory roles or Distribution lists.
4. Expand each of the three sections (**ID**, **Access**, **SAML**) and make sure **Group ID** is selected. This is the default. **Do not change it to sAMAccountName** for cloud-only Entra groups (sAMAccountName only works for groups synced from on-prem AD).
5. Click **Add**.

>

> **What PatchMon receives:** With this configuration, Entra ID sends groups as an array of GUIDs (the group Object IDs) in the `groups` claim of the ID token. You will paste those GUIDs (not group names) into PatchMon's Role Mapping table.

Optional but recommended: add standard user claims

Entra doesn't always include every OIDC-standard claim by default.

1. Still on **Token configuration**, click **+ Add optional claim**.
2. Token type: **ID**.
3. Tick `email`, `family_name`, `given_name`, `preferred_username`.
4. Click **Add**. If prompted to enable the Microsoft Graph `email` permission, accept.

Step 5: API Permissions

1. Open **API permissions** in the left menu.
2. You should already see `User.Read` listed under **Microsoft Graph**. That's enough. If it's missing, click **+ Add a permission → Microsoft Graph → Delegated permissions** and add `User.Read`, `openid`, `profile`, `email`.
3. Click **Grant admin consent for** at the top and confirm. Without admin consent, users will be prompted to consent individually on first login.

Step 6: Create Security Groups for Role Mapping

Decide which PatchMon roles you'll use. At minimum you probably want **Admin** and **User**. You can add more later.

For **each** role:

1. In Entra, go to **Identity → Groups → All groups**.
2. Click **+ New group**.
3. Fill in:
   - **Group type:** `Security`
   - **Group name:** e.g. `PatchMon Admins` (the name is for humans; PatchMon matches on Object ID)
   - **Membership type:** `Assigned` (simplest)
4. Add the users who should hold that role as **Members**.
5. Click **Create**.
6. After creation, open the group and **copy its Object ID** (a GUID like `11111111-2222-3333-4444-555555555555`) into your scratch note.

Repeat for each role you want to use.

Mapping table

PatchMon role	Entra group (example)	Where you'll paste the Object ID
Super Admin	`PatchMon SuperAdmins`	Role Mapping table → `superadmin` row
Admin	`PatchMon Admins`	Role Mapping table → `admin` row
Host Manager	`PatchMon Host Managers`	Role Mapping table → `host_manager` row
User	`PatchMon Users`	Role Mapping table → `user` row
Readonly	`PatchMon Readonly`	Role Mapping table → `readonly` row

> You only need to fill in the rows you use. Empty rows are ignored. Users who match none of the groups get the **Default (fallback)** role.

Part B: Configure PatchMon (Tab 1)

Go back to Tab 1: **Settings → OIDC / SSO**.

Step 7: Fill in the OAuth2 Configuration Section

Scroll to the **OAuth2 Configuration** panel and fill in the fields using the values from your scratch note:

Field in PatchMon	What to put in it
Issuer URL	`https://login.microsoftonline.com/<TENANT_ID>/v2.0`. Replace `<TENANT_ID>` with the Directory (tenant) ID from Step 2. The `/v2.0` suffix is required.
Client ID	The Application (client) ID from Step 2
Client Secret	Paste the client secret Value from Step 3, then click the **Save** button next to the field. The badge will change from "Not set" to "Set"
Callback URL	Read-only, already populated. This is the URL you registered in Entra in Step 2
Redirect URI (optional override)	Leave empty. Only use this if your PatchMon is behind a reverse proxy that presents a different public URL

Field in PatchMon	What to put in it
Scopes	Change the default `openid email profile groups` to **`openid email profile User.Read`**: remove the trailing `groups` and add `User.Read`. Entra rejects `groups` as an unknown scope. `User.Read` is required if you want PatchMon to fetch the user's Entra profile photo
Button Text	`Sign in with Microsoft` (or anything you like)

Click **Apply** at the bottom of the panel. You should see a toast saying **"OIDC settings saved"**.

> **Why no `groups` scope for Entra?** Other IdPs (Authentik, Keycloak) use a `groups` scope to request group claims. Entra does not. It uses the app's **Token configuration** instead (which you configured in Step 4). Including `groups` in the Scopes field will cause Entra to reject the authorisation request with an "invalid scope" error.
>
> **Why add `User.Read`?** PatchMon uses `User.Read` to call Microsoft Graph and fetch the signed-in user's profile photo. Without it, SSO still works, but Entra profile pictures cannot be imported.

Step 8: Configure the Toggles

At the top of the OIDC / SSO page there's a **Configuration** panel with five toggles. Recommended settings for Entra ID:

Toggle	Recommended	Why
Enable OIDC / SSO	**Leave OFF for now.** You'll turn it on in Step 10 after everything else is set	Flipping it on too early will expose a broken SSO button on the login page
Enforce HTTPS	**ON**	Entra will not work over plain HTTP anyway
Sync roles from IdP	**ON**	Required if you want Entra security groups to drive PatchMon roles
Disable local auth	**OFF** (for now)	Leave this off until you've confirmed SSO works. You can enable it later
Auto-create users	**ON**	Creates PatchMon accounts automatically on first login so you don't have to pre-provision users

No Save button is needed for the toggles at the top (except **Enable OIDC / SSO**, which saves immediately). The other four are applied when you click **Apply** in the OAuth2 Configuration panel.

Step 9: Fill in the Role Mapping Table

1. Scroll to **Role Mapping** and click the header to expand it.
2. You'll see a table with a **Default (fallback)** row and one row per PatchMon role.
3. For each role you created an Entra group for, paste the group's **Object ID** (from Step 6) into the **OIDC Mapped Role (IdP Group Name)** column.

PatchMon Role	Paste here
Default (fallback)	Leave as `user`, or change to `readonly` if you want unmatched users to have no write access
superadmin	Entra Object ID of `PatchMon SuperAdmins` (or leave blank if you don't want anyone promoted to superadmin via SSO)
admin	Entra Object ID of `PatchMon Admins`
host manager	Entra Object ID of `PatchMon Host Managers`
user	Entra Object ID of `PatchMon Users`
readonly	Entra Object ID of `PatchMon Readonly`

4. Scroll back up to the **OAuth2 Configuration** panel and click **Apply** to save the role mapping. (The role mapping fields are saved together with the OAuth2 fields by the Apply button.)

> **Important:** The label reads "IdP Group Name" but for Entra ID you must paste the group's **Object ID (GUID)**, not the display name. Entra sends GUIDs in the token, not names.

> **Amber warning:** If Sync Roles is on but the Superadmin row is empty, you'll see an amber warning. That is expected: it means no one will be promoted to superadmin via SSO. Existing local superadmins will keep their role. If that's what you want, ignore the warning.

Step 10: Turn On OIDC and Test

1. At the top of the page, flip **Enable OIDC / SSO** to **ON**. It saves immediately.
2. Open PatchMon in a **private/incognito browser window** (so you're not using your existing session).
3. You should see a **Sign in with Microsoft** button on the login page (or whatever text you set).

4. Click it. You'll be redirected to `login.microsoftonline.com`.
5. Sign in with an Entra account that's a member of one of your PatchMon groups.
6. You'll be redirected back and logged in.

First-login behaviour:

- A PatchMon account is created automatically. The username is derived from the email prefix (e.g. `alice@contoso.com` → `alice`).
- The role is determined by group membership; if no group matches, the **Default (fallback)** role is used.
- If **no admin exists yet in PatchMon**, the very first OIDC user is automatically promoted to **Super Admin** regardless of groups, so you cannot lock yourself out.

Optional: Enforce SSO Only (Disable Password Login)

Once you've confirmed at least one OIDC user has Admin or Super Admin:

1. Go back to **Settings → OIDC / SSO**.
2. Turn **Disable local auth** to **ON**.
3. Click **Apply** at the bottom of the OAuth2 Configuration panel.

The login page will now only show the **Sign in with Microsoft** button. Local username/password fields are hidden.

> **Safety:** PatchMon only enforces this flag if OIDC is **also** enabled *and* successfully initialised. If OIDC breaks for any reason, local login is automatically re-enabled so you're not locked out.

Troubleshooting

"OIDC is configured via .env" amber banner at the top

You'll see this if OIDC environment variables were set in `.env` before the UI was used. Click **Load from .env** to import those values into the database, then remove the `OIDC_*` lines from `.env` and

restart the server. From then on, everything is managed from the UI.

The "Sign in with Microsoft" button doesn't appear on the login page

The button only shows when OIDC is both **enabled** and **successfully initialised** at runtime. Most common causes:

- **Issuer URL is wrong:** it must end in `/v2.0`. Double-check for typos in the tenant GUID.
- **Client Secret is empty or wrong:** the label will say "Not set". Re-enter it and click **Save** next to the secret field.
- **PatchMon cannot reach `login.microsoftonline.com`:** an egress firewall or proxy is blocking it.

Check the server logs; search for `oidc`:

```
# Docker
docker compose logs patchmon-server | grep -i oidc


# Native systemd
journalctl -u <your-service-name> | grep -i oidc
```

`AADSTS50011: Reply URL does not match`

The redirect URI in Entra does not match the callback URL PatchMon is sending. Go to the Entra app's **Authentication** page and verify:

- Protocol is `https://`
- Host and port exactly match PatchMon's public URL
- Path is `/api/v1/auth/oidc/callback` with **no** trailing slash
- There are no hidden whitespace characters (paste into a plain editor to check)

If you're behind a reverse proxy and PatchMon is generating the wrong callback URL, fix the **Server URL** in **Settings → General** first. Do not use the "Redirect URI (optional override)" field unless you really know the proxy is presenting a different public URL.

`AADSTS70011: The provided value for scope ... is not valid`

Your Scopes field includes `groups`. Entra rejects unknown scopes. Change the Scopes field to:

```
openid email profile User.Read
```

Click **Apply**.

`AADSTS700016: Application with identifier ... was not found`

The **Client ID** field doesn't match the Application (client) ID in Entra. Copy it again from the app's **Overview** page and click **Apply**.

`AADSTS7000215: Invalid client secret provided`

The secret is wrong, was rotated, or has expired. Create a new one in Entra (**Certificates & secrets**), paste the new Value into the Client Secret field, and click **Save** next to the field.

Logged in but got the wrong role (or default role)

1. Make sure **Sync roles from IdP** toggle is ON.
2. Confirm you pasted the Entra group **Object ID (GUID)**, not the display name, into the Role Mapping table.
3. Check the server logs. PatchMon logs which groups it received:

   ```
   docker compose logs patchmon-server | grep -i "oidc groups"
   ```

4. If logs show `oidc no groups in token`, revisit Step 4 and make sure the groups claim was added under Token configuration with **Security groups → Group ID**.

Logged in but no profile photo appears

1. Make sure the **Scopes** field includes `User.Read`.
2. Confirm the Entra app has **Microsoft Graph → Delegated permission → User.Read** and that **admin consent** was granted.
3. Check whether the user actually has a profile photo set in Microsoft 365 / Entra.
4. Sign out and sign back in after changing scopes or permissions so PatchMon gets a fresh access token.

"Too many groups": user belongs to more than 200 groups

If a user is a member of 200+ groups in Entra, the token switches to a `_claim_names` overage indicator and omits the `groups` array. PatchMon does not currently follow the overage pointer.

Workaround: In Entra's **Token configuration → Edit groups claim**, select **Groups assigned to the application**. This limits the claim to groups explicitly assigned to the PatchMon app, which almost always keeps the total well under 200.

"Session Expired" after clicking the SSO button

The state cookie has a 10-minute TTL by default. If users take too long on the Microsoft login page (MFA, password reset), it expires. They just need to click the SSO button again and complete the login faster. If this happens often, the TTL is configurable via `OIDC_SESSION_TTL` in `.env` (this one is not yet in the UI).

Quick Reference: Where Each Value Comes From

PatchMon UI field	Where to find it in Entra
Issuer URL	`https://login.microsoftonline.com/<Directory (tenant) ID>/v2.0`. Tenant ID is on the Entra app's **Overview** page
Client ID	Entra app **Overview** → **Application (client) ID**
Client Secret	Entra app → **Certificates & secrets** → client secret **Value** (shown once, at creation time)
Callback URL	Already filled in by PatchMon. Copy it **to** Entra, not from it
Scopes	`openid email profile User.Read` (no `groups`)
Role Mapping → each row	Entra → **Groups → All groups → → Overview → Object ID**

Two Factor Authentication

PatchMon supports time-based one-time password (TOTP) two-factor authentication (2FA, sometimes called MFA) on top of the normal username / password login. Once enabled on a user's account, every sign-in asks for a 6-digit code from an authenticator app, or a one-time backup code.

This page covers enabling 2FA per user, using backup codes, the "Remember Me" trusted-device feature, and how admins recover an account if the user loses their authenticator.

> **Related pages:**
>
> - Users, Roles and RBAC: manage user accounts
> - Setting Up OIDC / Single Sign-On: delegate authentication to an external IdP
> - PatchMon Environment Variables Reference: the full env-var list

---

Scope and limitations

- 2FA is **opt-in per user**. Each user decides whether to turn it on from their own profile.
- 2FA is **not available for OIDC accounts.** If a user signs in via OIDC / SSO, their IdP is responsible for MFA. The PatchMon TFA tab is hidden on the profile page for OIDC-only accounts, and the setup endpoint refuses with "*MFA is managed by your OIDC provider*".
- There is **no global "require 2FA for all users"** flag in the current release. Administrators cannot enforce 2FA for every account from the UI or an environment variable. If you need enforced 2FA, drive authentication through an OIDC provider that enforces MFA (e.g. Authentik, Entra ID) and set `OIDC_DISABLE_LOCAL_AUTH=true`.
- The first-time setup wizard offers new admins the option to set up 2FA during initial account creation (Step 2 of the wizard). This is voluntary and can be skipped.

Enabling 2FA on Your Account

Each user enables 2FA themselves from their profile. Admins cannot enable it on behalf of another user.

1. Sign in to PatchMon with your username and password.
2. Click your avatar (top-right) → **Profile**.
3. Open the **Multi-Factor Authentication** tab.
4. Click **Enable TFA**.
5. A QR code appears. Scan it with your authenticator app of choice. Known-good options:
   - **Authy**
   - **Google Authenticator**
   - **1Password**
   - **Bitwarden**
   - **Microsoft Authenticator**
   - **Duo Mobile**
6. If you can't scan the QR code (shared device, desktop-only app), copy the **Manual Entry Key** instead and paste it into your authenticator.
7. Click **Continue to Verification**.
8. Enter the current 6-digit code from your authenticator app.
9. Click **Verify & Enable**.

You are now shown a one-time list of **backup codes** (see next section). Save them before clicking **Done**.

From now on, every password-based login will prompt for a 6-digit verification code after the password step.

Backup codes: save these

After enabling 2FA, PatchMon generates a batch of single-use backup codes. These let you sign in if you lose access to your authenticator app (lost phone, wiped device, etc.).

- **Each code can be used exactly once.** Once used, it is consumed and cannot be reused.
- **They are shown only once**, in plaintext, immediately after setup or regeneration. PatchMon stores them as bcrypt hashes in the database; neither you nor an admin can recover the plaintext later.
- **Treat them like a second password.** Store them in a password manager, or print them and lock them away.
- Click **Download Codes** to save a plain-text file for offline storage.

Regenerating backup codes

If you think your backup codes have leaked, or you've used most of them:

1. Go to **Profile → Multi-Factor Authentication**.
2. Scroll to the **Backup Codes** panel.
3. Click **Regenerate Codes**.

4. A new set of codes is generated and shown. The old set is immediately invalidated.

Using a backup code

On the 2FA prompt at login, you enter backup codes in the **same field** as TOTP codes. There is no separate "use a backup code" button. PatchMon tries the code as a TOTP first; if that fails, it checks whether it matches one of the stored backup-code hashes. If it matches, that backup code is consumed (removed from the stored list) and you are logged in.

Typical workflow if you've lost your phone:

1. At the login page, enter your username and password as usual.
2. On the "Two-Factor Authentication" screen, type one of your backup codes in the **Verification Code** field.
3. Click **Verify**.

The code is spent. Your next login cannot use the same backup code again.

"Remember Me": Trusted Devices

When you enter your 2FA code, there's a **Remember me on this computer (skip TFA for 30 days)** checkbox. If ticked, PatchMon plants a long-lived, HttpOnly `patchmon_device_trust` cookie on that browser and records a hashed trust token in the database.

On subsequent logins from the same browser:

- You still enter your password.
- PatchMon sees the trust cookie, matches it to the database record, confirms the record belongs to you and hasn't expired, and skips the 2FA prompt.
- A `last_used_at` timestamp on the trust record is bumped each time it's used, so you can see when each remembered device last signed in.

How the trust is keyed

The trust cookie is keyed only on **(user ID, cookie hash)**. It is deliberately **not** bound to IP address or user agent, so:

- Roaming between Wi-Fi, mobile hotspot, and office network does not invalidate the trust.
- Updating your browser does not invalidate the trust.
- Copying the cookie to a different browser on a different machine **would** bypass 2FA for that user (standard web cookie security model). Protect your browser profile accordingly.

Trust lifetime

The default lifetime is **30 days**, configurable server-wide via the `TFA_REMEMBER_ME_EXPIRES_IN` environment variable. Accepts duration strings such as `7d`, `30d`, `90d`. See PatchMon Environment Variables Reference for the full list.

There is a hard cap on how many trusted devices a single user can accumulate, controlled by `TFA_MAX_REMEMBER_SESSIONS` (default `5`). When a sixth device is trusted, the oldest existing trust is removed automatically.

Reviewing your trusted devices

1. Go to **Profile → Trusted Devices**.
2. You'll see a list with, for each device:
   - **Label** (best-effort device name derived from the user agent)
   - **User agent**
   - **IP address** at the time it was last used
   - **Created** / **Last used** / **Expires** timestamps
   - A **This device** badge next to the one you're currently logged in from

Revoking a trusted device

To stop a specific device skipping 2FA (for example, an old laptop you're decommissioning):

1. **Profile → Trusted Devices**.
2. Find the device in the list and click **Revoke**.
3. Confirm.

If the device you're revoking is the **current** browser, its trust cookie is also cleared, so your next login from this browser will require 2FA again.

Revoking every trusted device

Click **Forget all trusted devices** at the top of the panel. This:

- Removes every trust record for your account.
- Clears the trust cookie on the current browser.
- Forces a full 2FA prompt on every device next time you sign in.

Use this after a suspected account compromise or after losing a device.

Disabling 2FA

To turn 2FA back off on your own account:

1. **Profile → Multi-Factor Authentication**.
2. Click **Disable TFA**.
3. Enter your password to confirm.
4. Click **Disable TFA**.

Side effects:

- The TOTP secret is wiped from the database.
- All existing backup codes are invalidated.
- **All of your trusted devices are revoked.** This is intentional. The only purpose of a trust record is to skip 2FA, so with 2FA off they serve no purpose. If you later re-enable 2FA, old trust cookies will not be resurrected and every device must confirm 2FA again.

> You cannot disable 2FA on an OIDC-only account. The API rejects the request with "*Cannot disable TFA for accounts without a password*". This is because disabling 2FA requires password confirmation, and OIDC-only accounts have no password set.

Failed Attempts and Lockout

To prevent brute-forcing the 6-digit code space, the verify-2FA endpoint is rate-limited per user.

Env var	Default	What it does
`MAX_TFA_ATTEMPTS`	`5`	Consecutive wrong codes allowed before a lockout
`TFA_LOCKOUT_DURATION_MINUTES`	`30`	How long the lockout lasts

After the cap is hit, the endpoint returns HTTP `429 Too Many Requests` with the message "*Too many failed TFA attempts. Please try again later.*" Wait out the lockout, or ask an admin (see below).

Each failure also returns a `remainingAttempts` counter in the response, so the login UI can tell the user how many tries are left.

First-Time Wizard: Optional 2FA Setup

When you bring up a brand-new PatchMon instance and complete the setup wizard, **Step 2 (Multi-Factor Authentication)** offers two choices:

- **Setup MFA now:** scan a QR code and register an authenticator for the brand-new admin account before finishing the wizard. You'll also capture your backup codes.
- **Skip for now:** the admin account is created without 2FA. You can turn it on later from **Profile → Multi-Factor Authentication**.

There is no "enforce for everyone" option in the wizard. This decision is always per-user.

Admin Recovery: User Has Lost Their Authenticator

PatchMon does not have a dedicated "admin reset MFA" button. Recovery is handled through the standard account-recovery flow, which implicitly disables 2FA in a safe way:

Option A: User has a backup code

Ask them to sign in with a backup code (see Using a backup code). Once they're in, they can:

1. **Profile → Multi-Factor Authentication → Disable TFA** to remove the old authenticator secret entirely, and then re-enable with the new phone.
2. Or **Regenerate Codes** to get a fresh set of backup codes without touching the authenticator.

Option B: User has no backup codes and no authenticator

An administrator must reset the account:

1. Sign in as a user with `can_manage_users` (admin, superadmin, or any custom role with that permission).
2. Go to **Settings → Users**.
3. Find the affected user and click **Reset Password**.
4. Set a new password and communicate it over an out-of-band secure channel.

> **Password reset alone does not disable 2FA.** The user will still be prompted for a TOTP or backup code after their first login with the new password.

If the user still cannot produce a code, you have two further options:

- **Deactivate, then reactivate the account.** Edit the user, untick **Active**, save (this also wipes their trusted devices), then tick **Active** again. 2FA is still enabled on the account, so this alone does not solve the missing-authenticator problem.
- **Delete and re-create the user** as a last resort. You lose the user's ID, notification preferences, and any artefacts keyed to their account, so prefer the backup-code route wherever possible.

> **Feature gap:** A "wipe 2FA on another user" admin action is on the roadmap. If you hit this frequently, consider moving your deployment to OIDC / SSO so that MFA is managed by the IdP (see Setting Up OIDC / Single Sign-On).

Direct database workaround (self-hosted only)

If you are self-hosting and absolutely need to clear 2FA on a user without backup codes, a DBA can clear the user's `tfa_enabled`, `tfa_secret` and `tfa_backup_codes` columns directly in the `users` table, then force a password reset from the UI. This is a last resort. Make a backup first, and never do this on PatchMon Cloud (where direct database access is not available).

```
-- Replace 'alice' with the affected username. Make a backup first.
UPDATE users
SET tfa_enabled = false,
    tfa_secret = NULL,
    tfa_backup_codes = NULL
WHERE username = 'alice';
```

After running this the user can sign in with just a password; they should immediately re-enrol in 2FA from their profile.

Environment Variables Reference

All of these are read once at server start. Changes require a restart to take effect. The full table lives in PatchMon Environment Variables Reference; reproduced here for convenience:

Variable	Default	Description
`MAX_TFA_ATTEMPTS`	`5`	Consecutive wrong 2FA codes before the account is temporarily locked
`TFA_LOCKOUT_DURATION_MINUTES`	`30`	How long a 2FA lockout lasts
`TFA_REMEMBER_ME_EXPIRES_IN`	`30d`	How long a "Remember me" trusted-device record is valid. Accepts `7d`, `30d`, `90d`, etc.
`TFA_MAX_REMEMBER_SESSIONS`	`5`	Maximum number of trusted devices per user; the oldest is evicted when the limit is reached

---

Troubleshooting

"Invalid verification code" when I know the code is correct

1. **Clock skew.** TOTP codes are time-based. If your phone's clock is more than ~30 seconds out of sync with the server, codes will be rejected. Enable automatic date/time on your phone. PatchMon already tolerates a small drift window server-side, but not more than that.
2. **Using a used code.** TOTP codes roll every 30 seconds. If you paste a stale code from 60+ seconds ago it will fail. Wait for a fresh code.
3. **Used backup code.** Backup codes are single-use. If you've already used one, try a different one.

"Too many failed TFA attempts"

You've hit `MAX_TFA_ATTEMPTS`. Wait `TFA_LOCKOUT_DURATION_MINUTES` (default 30) and try again. There is no admin "unlock" button; the lockout key in Redis expires automatically. Self-hosters can flush the key by restarting Redis.

I ticked "Remember me" but I'm still being asked for 2FA

Three likely causes:

- The trust record has expired. The default lifetime is 30 days; check `TFA_REMEMBER_ME_EXPIRES_IN` on your server.

- You're signing in on a different browser, or in a private / incognito window, which doesn't have the cookie.
- A password reset was performed on your account. Password resets (whether self-service or admin-initiated) revoke every trusted device as part of the security response. You'll need to tick **Remember me** again on the next 2FA prompt.

My MFA tab is missing on the profile page

You signed in via OIDC. PatchMon defers MFA to your IdP in that case. Enable MFA in your IdP (Entra ID, Authentik, Keycloak, etc.) if you want it.

I regenerated backup codes but the old ones still work

The old codes are invalidated at the same moment the new batch is displayed. If a stale code still seems to work, make sure you're looking at the right account. Backup codes are not user-transferable.

Users, Roles and RBAC

Users, Roles and RBAC

PatchMon uses role-based access control (RBAC) to decide who can see and do what inside the application. Every user has exactly one role, and every role is a collection of permissions. This page covers the built-in roles, the full permission list, and how to manage users and roles from the Settings UI.

> **Related pages:**
>
> - Setting Up OIDC / Single Sign-On: authenticate users against an external IdP
> - Setting Up Microsoft Azure Entra ID (SSO) with PatchMon: Entra-specific walkthrough
> - Two-Factor Authentication: per-user TOTP and trusted devices

---

The Built-In Roles

PatchMon ships with five roles. You see these in **Settings → Users** (in the **Role** dropdown) and in **Settings → Roles** (as the matrix columns).

Role	Default Permissions	Typical Use
Super Admin (`superadmin`)	Everything, including managing other superadmins	The very first user, or dedicated platform owners
Admin (`admin`)	Everything except managing other superadmins	Day-to-day platform administrators
Host Manager (`host_manager`)	Monitoring + host/infrastructure management + operations (patching, compliance, alerts, automation, remote access)	NOC / Ops engineers
User (`user`)	Monitoring + data export	Engineers who need to look but not break
Readonly (`readonly`)	Monitoring only	Auditors, read-only dashboards, management

Two important rules about built-ins:

- **Cannot be deleted.** `superadmin`, `admin`, `host_manager`, `user` and `readonly` are always present. The **Delete** button does not appear for them.
- **The core three cannot have their permissions edited.** `superadmin`, `admin` and `user` are *locked*: their permission matrix is hardcoded and the **Edit** button is disabled. `host_manager` and `readonly` can still be edited if you want to tune them.

> **First user is always Super Admin.** When PatchMon is first installed and has no users, the setup wizard creates the initial account as `superadmin`, regardless of what role you type. If OIDC is configured for auto-create before first boot, the very first OIDC login is also promoted to `superadmin` automatically so you cannot lock yourself out.

The Full Permission List

Permissions are grouped into four risk tiers. The colour you see in the **Roles** matrix corresponds to this risk level.

Monitoring & Visibility (Low risk)

Read-only access to dashboards, hosts, packages, reports, and logs.

Permission key	Label	What it lets the user do
`can_view_dashboard`	View Dashboard	View the main dashboard and its stat panels
`can_view_hosts`	View Hosts	See the host list, host detail pages, and connection status
`can_view_packages`	View Packages	See the package inventory across all hosts
`can_view_reports`	View Reports	See compliance scan results and alert reports
`can_view_notification_logs`	View Notification Logs	See notification delivery history and status

Host & Infrastructure (Medium risk)

Create, modify and delete hosts, packages, and containers.

Permission key	Label	What it lets the user do
`can_manage_hosts`	Manage Hosts	Create / edit / delete hosts, host groups, repositories and integrations
`can_manage_packages`	Manage Packages	Edit package inventory and metadata
`can_manage_docker`	Manage Docker	Delete Docker containers, images, volumes and networks

Operations (Medium-High risk)

Day-to-day NOC tasks.

Permission key	Label	What it lets the user do
`can_manage_patching`	Manage Patching	Trigger patches, approve patch runs, manage policies
`can_manage_compliance`	Manage Compliance	Trigger compliance scans, remediate findings, install scanners
`can_manage_alerts`	Manage Alerts	Assign, delete and bulk-action alerts
`can_manage_automation`	Manage Automation	Trigger and manage automation jobs
`can_use_remote_access`	Remote Access	Open SSH and RDP terminals against managed hosts

Administration (High risk)

Organisation-wide control.

Permission key	Label	What it lets the user do
`can_view_users`	View Users	See the user list and account details
`can_manage_users`	Manage Users	Create, edit and delete user accounts
`can_manage_superusers`	Manage Superusers	Manage `superadmin` accounts and elevated privileges
`can_manage_settings`	Manage Settings	System configuration, OIDC / SSO, AI, alert config, enrollment tokens
`can_manage_notifications`	Manage Notifications	Configure notification destinations and routing rules
`can_export_data`	Export Data	Download and export data and reports

> **Billing:** On PatchMon Cloud there is also a `can_manage_billing` permission that governs access to the Billing page. On self-hosted instances this permission exists in the schema but the Billing page is not enabled by default.

Viewing the Role Matrix

1. Sign in as a user with `can_manage_settings`.
2. Go to **Settings → Roles**.
3. You'll see a matrix: rows are permissions (grouped by tier), columns are roles. A green tick means the role has that permission.

Each column header also shows an `n/N` counter showing the number of permissions that role currently holds out of the total 20.

Creating a Custom Role

Custom roles let you tailor the permission set beyond the built-in five.

> **Availability:** The **Add Role** button is only shown when the `rbac_custom` module is enabled on your PatchMon deployment. On self-hosted installs this module is typically enabled by default; on PatchMon Cloud it depends on your plan. If you don't see **Add Role** and the URL `https://patchmon.example.com/settings/roles` shows a "Not Available" screen, the module isn't enabled on your plan.

To create one:

1. Go to **Settings → Roles**.
2. Click **Add Role** in the top-right.
3. Fill in the modal:
   - **Role Name:** lowercase, underscores instead of spaces. Examples: `host_manager`, `compliance_auditor`, `noc_operator`. This is the internal key; it cannot be renamed later.
   - **Preset** (optional): four quick-start presets are available:
     - **Read Only:** just the Monitoring & Visibility group
     - **Operator:** everything except the Administration group
     - **Admin:** every permission
     - **Clear All:** start from zero

- **Permissions:** tick / untick individual permissions, or use the **Select all / Deselect all** shortcut on each group header.

4. Watch the counter at the bottom (`n/20 permissions selected`) as a sanity check.
5. Click **Create Role**.

The new role appears as a new column in the matrix and is selectable when creating or editing users.

Editing a Custom Role

1. In the matrix, click the pencil icon in the column header of the role you want to edit.
2. An editor panel opens below the matrix with all permissions listed.
3. Tick / untick as needed, then click **Save**.

Changes take effect immediately. Any session held by a user with that role has its in-memory permissions refreshed on their next request.

Deleting a Custom Role

You can only delete a role that is **not assigned to any user**. If any user holds that role, the delete endpoint rejects the request with "Cannot delete role: users are assigned to it". Reassign those users to a different role first (see Editing a Role for an Existing User).

To delete:

1. Click the pencil in the role's column header to open the editor panel.
2. Click **Delete** (appears only for non-built-in roles).
3. Confirm.

Creating Users

Go to **Settings → Users** and click **Add User** in the top-right.

Field	Notes
Username	Minimum 3 characters. Lowercase recommended
Email	Must be a valid email. Used for OIDC account linking and email alerts
First Name / Last Name	Optional
Password	Must satisfy the active password policy (configured under **Settings → Server Config → Security**)

Field	Notes
Role	Choose from built-in or custom roles

Click **Add User**. The account is created immediately and can sign in straight away.

> **Role escalation protection:** You cannot create a user with a role that's more privileged than your own. Only `superadmin` users can create new `admin` or `superadmin` accounts. Non-superadmin accounts that hold the `can_manage_superusers` permission can also create and manage `superadmin` accounts.

Self-Service Sign-Up

PatchMon can also let users register themselves rather than having an admin invite them.

1. Go to **Settings → Users**.
2. Scroll to **User Registration Settings**.
3. Tick **Enable User Self-Registration**.
4. Pick a **Default Role for New Users**: the role that self-registered accounts are assigned.
5. Click **Save Settings**.

A sign-up link now appears on the login page. Anyone who can reach the login page can create an account.

> **Security warning:** Only enable self-registration on internal or private-network deployments. If your PatchMon is internet-facing, leave it off and invite users manually, or front it with OIDC SSO (which lets your IdP decide who can log in).

Editing a Role for an Existing User

1. Go to **Settings → Users**.
2. Find the user in the table and click the **Edit** (pencil) icon.
3. Change **Role** in the dropdown and click **Save**.

Important side effects:

- **Sessions are revoked.** When a user's role changes, all of their existing JWT sessions are invalidated on the server. They must sign in again. This ensures the old role's privileges

cannot be replayed from an existing browser tab.
- **You cannot change your own role.** The API rejects a self-role change with "Cannot change your own role". This is a deliberate safety net: two admins must cooperate to demote each other.
- **You cannot promote a user above yourself.** An `admin` cannot promote a user to `superadmin`. Only a `superadmin` can create or promote to `superadmin`, and likewise only `superadmin` can assign the `admin` role.

Resetting a User's Password

1. In the users table, click the **Reset** (key) icon on that user's row.
2. Enter a new password.
3. Click **Reset Password**.

After a reset, all of that user's sessions and trusted-device records are revoked. This is the standard post-compromise response. The user must sign in with the new password on every device.

> You cannot reset the password of an inactive user. Reactivate them first.

Disabling (Deactivating) a User

Disabling is the safer alternative to deletion. The user record, their history, and their audit trail are preserved, but they cannot log in.

1. Go to **Settings → Users**.
2. Click the **Edit** icon on the user you want to disable.
3. Untick the **Active** checkbox.
4. Click **Save**.

Effects:

- All their sessions are revoked immediately.
- All their trusted devices are revoked (so re-activating them later cannot reuse a "remember this device" cookie that predates the deactivation window).
- The user's row is shown with a red **Inactive** badge in the users table.

To re-enable: edit and tick **Active** again.

Deleting a User

Deletion is permanent and removes the user record and their associated dashboard preferences, sessions, trusted devices and notification preferences.

1. Click the **Delete** (trash) icon on the user's row.
2. Confirm.

Restrictions:

- You cannot delete your own account.
- You cannot delete the last `superadmin` (the API refuses).
- You cannot delete the last `admin` if there are no `superadmin` users (ensures at least one admin always exists).
- You cannot delete a user who holds a role that's more privileged than yours.

How Permissions Are Evaluated

- **Admin and Super Admin** always have every permission, even if the `role_permissions` table says otherwise. The middleware short-circuits their permission checks. This is a safety net: if someone mis-edits the `admin` row (which shouldn't be possible via the UI, but could happen via direct database access), admins don't get locked out.
- **Every other role** (built-in or custom) has its permissions read from the database at each request. Changes made in **Settings → Roles** take effect on the user's next API call; no restart required.
- **Role hierarchy for user management** is enforced separately from the permissions above:
  - `superadmin` → rank 100
  - `admin` → rank 90
  - `host_manager` → rank 50
  - custom roles → rank 30 (mid-tier)
  - `user` → rank 20
  - `readonly` → rank 10

You can only modify, delete, or reset the password of users whose role rank is less than or equal to your own. This is distinct from the permission checks. Even if a custom role were granted `can_manage_users`, its holder still could not touch `admin` or `superadmin` accounts unless they additionally had `can_manage_superusers`.

When OIDC Role Sync Is Enabled

If **Settings → OIDC / SSO → Sync roles from IdP** is on, PatchMon stops letting admins manage users and roles from the UI. Instead:

- The **Add User** and **Add Role** buttons disappear.
- The Users tab shows a read-only list.
- The Roles tab shows a banner reminding you that group membership in your IdP drives role assignment via environment variables: `OIDC_SUPERADMIN_GROUP`, `OIDC_ADMIN_GROUP`, `OIDC_HOST_MANAGER_GROUP`, `OIDC_USER_GROUP`, `OIDC_READONLY_GROUP`.
- Users' roles are re-evaluated on every login based on their current IdP group membership.

If you want to use OIDC for authentication but still manage roles locally in PatchMon, leave **Sync roles from IdP** off. See Setting Up OIDC / Single Sign-On for the full toggle reference.

Troubleshooting

"You do not have permission to assign the role: admin"

Only a `superadmin` can create or promote users to `admin` or `superadmin`. If you're an `admin` and try to promote someone to `admin`, the API refuses. Ask a superadmin to do it.

"Cannot modify built-in role permissions"

The `superadmin`, `admin` and `user` rows are locked against permission edits. If you need a role with tweaked permissions, create a custom role based on a preset and assign users to that instead.

"Cannot delete role: users are assigned to it"

Before a role can be deleted, reassign every user who holds it. Use **Settings → Users → Edit** to change each user's role, then try the delete again.

"Cannot delete the last superadmin user" / "Cannot delete the last admin user"

At least one `superadmin` must always exist. If there are no superadmins at all, at least one `admin` must exist. Create a replacement first (and sign in as them to confirm the login works) before deleting the final one.

User's old role is still in effect after I changed it

Changing a role revokes all existing sessions, but the user's browser may still hold an old JWT cookie that hasn't been rejected yet. Ask them to refresh the page or sign out and back in; the server will reject the stale token and redirect them to login.

"Add User" / "Add Role" button is missing

Three possible causes:

1. **Your role doesn't have `can_manage_settings` or `can_view_users`.** Check `/settings/users`: if the page is empty or you get a Forbidden, your role lacks the view permission.
2. **OIDC role sync is on.** See When OIDC Role Sync Is Enabled.
3. **The `rbac_custom` module is not enabled.** This only affects the **Add Role** button on the Roles tab. Custom role creation is a gated feature. The **Add User** button on the Users tab is always available when the other two conditions are met.