matomo

Security Checklist

For Matomo Cloud and On-Premise

Take control of your data, without compromise.

Updated 2026

How to use this checklist: Go through each section and tick off items as you complete them. Items are grouped by deployment type: steps that apply to **both** Cloud and On-Premise, then **ON-PREMISE only**. Start with "Both", these are the fundamentals every Matomo instance needs.

BOTH Applies to Cloud & On-Premise   **ON-PREMISE** Self-hosted only

Difficulty: ●○○ Easy   ●●○ Medium   ●●● Advanced

A — Cloud & On-Premise

These steps apply to every Matomo deployment. Complete them first.

1 Apply the principle of least privilege

●○○ **BOTH**

- □ Give each person the **minimum role needed**: View > Write > Admin > Super User.
- □ Keep **Super User accounts to a strict minimum** (ideally 2 max).
- □ **Restrict allowed email domains** for user login (System > General Settings).
- □ Enable **inactive user alerts** and remove accounts no longer needed.

*Roles: **View** = read-only reports · **Write** = create goals, segments, heatmaps · **Admin** = manage site settings, GDPR tools · **Super User** = full system control*

Ref: View permission · Admin permission · Secure accounts

2 Enable two-factor authentication (2FA)

●○○ **BOTH**

- □ Enable 2FA for **all user accounts** via Administration > Personal > Security.
- □ Super Users: **enforce 2FA system-wide** for every account.
- □ **Back up recovery codes** in a secure location (password manager).
- □ Enforce **strong, unique passwords** for every account.

Ref: What is 2FA? · Enforce strong passwords · Security guide

3 Verify anti-intrusion settings

●○○ **BOTH**

- □ Check the trusted hosts list (Administration > General Settings) to prevent host injection.
- □ Verify **brute force protection** is enabled (on by default).
- □ Whitelist known admin IPs to prevent accidental lockouts.
- □ **Use Single Sign-On (SSO)** to centralise authentication and secure Matomo access.

Ref: Brute force FAQ · Configure for security · Single Sign-on (SSO)

4 Secure your API tokens ●●○ BOTH

- □ Treat every **token_auth** as a password. Never share it.
- □ Generate a **separate token** for each application or integration.
- □ Always send tokens in a secure way (POST parameter or Bearer Header).
- □ Use **expiring credentials** when the integration supports it and **audit all API tokens regularly** to revoke those no longer in use.
- □ Store tokens in a **password manager** immediately upon creation.

Ref: Token auth security

B — On-Premise only

Self-hosted Matomo requires additional server-level hardening. These steps are your responsibility.

5 Keep Matomo up to date ●●○ ON-PREMISE

- □ Subscribe to the **Matomo Changelog**, our **Newsletter** and apply updates promptly.
- □ After each update, check **Administration › Diagnostics › System check**.
- □ Review and **update all plugins** and themes regularly.

Ref: Update Matomo · Changelog

6 Enforce HTTPS/SSL everywhere ●●○ ON-PREMISE

- □ Always access Matomo over **https://**. Enable **force_ssl = 1** in your config file.
- □ If behind a reverse proxy, also set **assume_secure_protocol = 1**.
- □ Enable **secure cookie flags** for all tracking cookies.
- □ Use **SFTP or SSH** for file transfers. Never unencrypted FTP.

Ref: Force SSL FAQ · Secure cookies · Secure Matomo guide

7 Secure your database ●●○ ON-PREMISE

- □ Use a **dedicated database** for Matomo. Never share with a CMS or other app.
- □ Create a **dedicated MySQL user** with unique credentials for Matomo.
- □ **Block all external network access** to the database server.
- □ Schedule **regular backups** of the database + **config/config.ini.php**.

Ref: Configure for security · Backup MySQL

8 Monitor and audit ●●○ ON-PREMISE

- □ Install the **SecurityInfo plugin** (free) to test your server security posture.
- □ Install the **Activity Log plugin** (premium) to audit all account actions.
- □ Review **Administration › Diagnostics › System check** regularly.

Ref: Security overview · Configure for security

9 Keep your server stack updated

●●○ ON-PREMISE

- □ Update **PHP** to the latest supported version.
- □ Update **MySQL/MariaDB** with the latest patches.
- □ Update **Apache/Nginx** to the latest stable release.
- □ Clean up leftover files: **./console diagnostics:unexpected-files**

Ref: Update Matomo

10 Secure auto-archiving (cron)

●●● ON-PREMISE

- □ Disable browser-triggered archiving: **browser_archiving_disabled_enforce = 1**
- □ Run the cron job as the web server user: **sudo -u www-data php console core:archive**
- □ Prefer core:archive (no token needed). For URL-based archiving, always pass token_auth via POST, not in the URL.
- □ Store cron log files **outside the web root**.

Ref: Auto-archiving setup · Monitor cron errors

11 Block access to private directories

●●○ ON-PREMISE

- □ **Apache:** enable **AllowOverride All** then run **./console core:create-security-files**
- □ **Nginx:** use the **official Matomo Nginx config** from GitHub.
- □ Verify these are **not publicly accessible**: /.git/ /core/ /config/ /lang/ /tmp/

Ref: Security config

Resources and further reading

Matomo security overview

Cloud security guide

Update Matomo

On-Premise hardening guide

Privacy features

Backup best practices

Matomo is ISO/IEC 27001:2022 certified. Your data is in safe hands.