scandiweb

Magento security checklist

31 checks in 10 areas: what to check, how to verify it, and how often

Checked against Adobe and PCI SSC sources on November 12, 2026

(Adobe) and (PCI DSS): the interval comes from that source. (our advice): scandiweb's recommendation.

#	Check	How to verify	How often	Done	Owner and date
1. Patch level and supported version					
1	Your Magento version still gets security fixes	Read the version in the bottom-right corner of any Admin page, or run `bin/magento --version`. Compare it with the end dates in Adobe's lifecycle policy.	Every quarter, and six months before your version's end date (our advice)	☐	
2	The latest -p release and every monthly patch since are installed	Run `php vendor/bin/patch-status`. The missing_patches list is empty and every CVE shows PROTECTED. On Adobe Commerce on cloud infrastructure, also check the Cloud Patches version.	Monthly, after Patch Tuesday, the second Tuesday of the month (Adobe)	☐	
3	Emergency hotfixes are applied, and a named person hears about them	A team inbox is subscribed to the Adobe Security Notification Service. Every hotfix bulletin since your last -p release matches a deployed patch.	The day a hotfix bulletin arrives (our advice)	☐	
2. Admin access					
4	Two-factor authentication is on for every admin user	Run `bin/magento module:status Magento_TwoFactorAuth` and confirm the module is enabled. Check the required providers at Stores > Settings > Configuration > Security > 2FA. With Adobe IMS sign-in, check 2FA in your Adobe organization instead.	Monthly (our advice)	☐	
5	Every admin account belongs to one current person, with only the role they need	System > Permissions > All Users: no shared or generic logins, no accounts of people who have left, and an expiration date on temporary users. Open each user's User Role tab.	Quarterly, and the day someone leaves (our advice)	☐	
6	Admin security settings have not drifted	Stores > Settings > Configuration > Advanced > Admin > Security: Admin Account Sharing is No, a lockout after failed logins is set, and Password Lifetime is set. Run `bin/magento info:adminuri` to confirm the Admin path is not the default.	Quarterly (our advice). Adobe recommends changing passwords at least every 90 days.	☐	
3. Extensions and custom code					
7	Every installed module is known and still needed	Run `bin/magento module:status --enabled` and `composer show`. Match the output to a list your team approved, and remove what nobody can name.	Monthly, with the patch slot (our advice)	☐	

#	Check	How to verify	How often	Done	Owner and date
8	No installed package has a known flaw or a pending security update	Run `composer audit` and `composer outdated`. For extensions that are not on Packagist, check the vendor's release notes.	Monthly, and when a vendor announces a fix (our advice)	☐	
9	Production code matches version control	A developer compares the code on the production server with the repository. Every file that differs is explained or removed.	Monthly (our advice), and at once after a suspected break-in (Adobe)	☐	
4. Payment page scripts					
10	Content Security Policy is enforced and Magento's scripts carry SRI hashes on checkout	Open the checkout with the browser's developer tools. The page response has a Content-Security-Policy header, not only Content-Security-Policy-Report-Only, and Magento's script tags show an integrity attribute. Restrict mode on payment pages is the default since 2.4.7.	After every deployment (our advice)	☐	
11	CSP violation reports reach someone	A Report URI is set at Stores > Settings > Configuration > Security > Content Security Policy, and a named person reads what arrives there.	Weekly (our advice)	☐	
12	Every checkout script is listed, approved, and has a written reason (PCI DSS 6.4.3)	Compare the scripts the checkout loads (developer tools, Network tab, JS filter) with your script inventory. Each script has an owner and a business or technical reason.	Whenever a script is added or changed (PCI DSS)	☐	
13	A tamper alert covers the checkout's headers and scripts (PCI DSS 11.6.1)	A change- and tamper-detection tool checks the headers and scripts the browser receives on the payment page and alerts a named person. Make one test change and confirm the alert arrives.	At least weekly, or at the interval in your targeted risk analysis (PCI DSS). Test the alert quarterly (our advice).	☐	
5. Web application firewall and bot protection					
14	A web application firewall filters traffic to the store	Adobe Commerce on cloud infrastructure: the Fastly WAF is active on production. Self-hosted: your host or CDN shows the active WAF policy and last week's blocked requests.	Monthly, and the day an exploited Magento flaw is announced (our advice)	☐	
15	Bots are rate-limited on login, checkout, and coupon codes	Your CDN or WAF shows rate-limit rules for those pages. On cloud, the Fastly WAF has no bot mitigation or rate limiting, so confirm Adobe's Advanced Security add-on or your own rules.	Quarterly (our advice)	☐	
16	reCAPTCHA protects admin sign-in and the forms bots target	Stores > Settings > Configuration > Security > Google reCAPTCHA Admin Panel (Admin Sign In, Forgot Password) and Google reCAPTCHA Storefront (Customer Login, Create New Customer Account, Checkout/Placing Order, Coupon Codes).	Quarterly, and after any admin lockout is fixed (our advice)	☐	
6. Malware scanning					
17	The Security Scan Tool runs on every domain, and someone reads the report	In your Commerce or Magento account, under Security Scan: every storefront domain is verified, a weekly scan is scheduled, and every Failed item in the latest report has an owner. Review anything marked Ignore.	Weekly (Adobe)	☐	

#	Check	How to verify	How often	Done	Owner and date
18	A server-side scanner checks files and the database	The scanner's latest report covers code, media folders, and database content such as CMS blocks, and its alerts reach a named person.	Daily (our advice)	☐	
19	No unknown code where skimmers hide	Content > Elements > Blocks, and Content > Design > Configuration for every store view (HTML Head > Scripts and Style Sheets, Footer > Miscellaneous HTML), hold only scripts your team added. pub/media holds no PHP files.	Weekly (our advice)	☐	
7. Encryption key, integration tokens, and AI agent access					
20	Every integration, including each AI agent, has an owner and only the access it needs	System > Extensions > Integrations: every Active integration has a named owner and Resource Access set to Custom. Delete the ones nobody uses, because an integration token lasts until it is revoked.	Monthly, and when a supplier or agent changes (our advice)	☐	
21	Token settings are still at their safe defaults	Stores > Settings > Configuration > Services > OAuth: "Allow OAuth Access Tokens to be used as standalone Bearer tokens" is No, and the admin token lifetime under Access Token Expiration has not been raised above the 4-hour default.	Quarterly (our advice)	☐	
22	The encryption key and credentials are rotated after any exposure, and keys stay out of code	Your change log shows when `bin/magento encryption:key:change` last ran (command line only since 2.4.8) and which credentials were rotated with it. A secret scan of the repository and of AI agent configuration files finds no keys.	After any incident or suspected exposure (Adobe), and the secret scan monthly (our advice)	☐	
8. Server and hosting					
23	The store runs a PHP version that both Adobe and PHP still support	A developer confirms the PHP version the store runs and compares it with Adobe's system requirements for your Magento version and PHP's own end dates. PHP 8.1 lost security fixes on December 31, 2025, and PHP 8.2 loses them on December 31, 2026.	Quarterly, and before each PHP end date (our advice)	☐	
24	Production runs in production mode	`bin/magento deploy:mode:show` returns production.	After every deployment (our advice)	☐	
25	Code folders are read-only and env.php is closed	Self-hosted: vendor, app/code, app/etc, pub/static, generated/code, generated/metadata and var/view_preprocessed have no write access in production, and app/etc/env.php has no permissions for other users. On cloud infrastructure, environments are read-only by design.	After every deployment (our advice)	☐	
26	No development leftovers are reachable	From outside your network, request /.git/HEAD, any phpinfo file, database dumps, and log files. Each returns 403 or 404.	Monthly (our advice)	☐	
9. Backups and disaster recovery					

#	Check	How to verify	How often	Done	Owner and date
27	Backups run and are stored away from the store's server	Self-hosted: your backup tool's log shows a recent successful database and media backup (Magento's built-in backup is deprecated). Cloud: Pro staging and production get automatic backups, and on Starter and integration environments `magento-cloud snapshot:list` shows your own.	Weekly look at the latest successful backup (our advice)	☐	
28	A restore works, and a written recovery plan names who does what	Restore the latest backup to a staging copy and place a test order. The disaster recovery plan names who restores and which credentials get rotated, and it notes how long the last restore took.	Restore test quarterly and before every upgrade, plan review yearly (our advice)	☐	
10. Logging and monitoring					
29	Admin actions are logged and reviewed	Adobe Commerce: System > Action Logs > Report, with Admin Actions Logging on. Magento Open Source has no Action Logs, so review the web server's access logs for Admin URLs instead.	Weekly (our advice)	☐	
30	No unknown or locked admin users	Compare System > Permissions > All Users with last week's list, and check System > Permissions > Locked Users for lockouts that may point to password guessing.	Weekly (our advice)	☐	
31	Alerts reach a named person, and logs go back far enough to trace an intrusion	Send a test alert from the scanner, the CSP reports and the tamper tool, and confirm a person received each one. Check the date of your oldest available log.	Quarterly (our advice)	☐	