Websites and CMS: who patches your site, and how fast?

This week, get every site's PHP patched for October's exploited flaws. This month, turn on automatic updates for core, plugins and themes, require MFA for every admin, plan the move off PHP 8.2 and Drupal 10, which lose support in December, and write down who patches and who restores the site when it is hacked.

Reviewed	Next review	Version	Written for
5 Nov 2026	**5 Dec 2026**	**1.1**	**Marketing, IT and security teams that run company websites and online shops, alone or with an agency or hoster**

Latest version
sheets.einblick.dev/websites-cms/

If you only do five things

1. **Turn on automatic updates for core and plugins.**
   Enable them for core and each plugin and theme[1]. In Patchstack's vendor report, heavily exploited WordPress flaws were first attacked a weighted median of 5 hours after disclosure[2].
2. **Delete every plugin you don't need.**
   Patchstack counted 11,334 new WordPress flaws in 2025: 91% in plugins, 46% unfixed at disclosure[2]. Install new ones only from the official directory or a well-known vendor[3].
3. **Put MFA on every admin account.**
   Give each person their own account, with company SSO or the CMS's own MFA. TYPO3 enforces it with `requireMfa`[4]; Wordfence's free WordPress plugin added passkeys in August 2026[5].
4. **Write down who patches what.**
   In agency and hosting contracts, name who updates the CMS, plugins and PHP, who restores, and how fast. German law makes businesses secure web systems against unauthorized access, within reason and considering the state of the art[7].
5. **Leave end-of-life versions before December.**
   PHP 8.2 loses security support on 31 December 2026[8] and Drupal 10 on 9 December[9]. Ask your hoster which PHP each site runs.

If the site is hacked

1. **Take it offline and keep the evidence.**
   Show a maintenance page, then copy the files, database and all logs, or have the hoster do it, before anything changes[10].
2. **Lock everyone out.**
   Change the passwords for CMS users, SFTP, the hosting panel and the database. In WordPress, replace the secret keys in `wp-config.php` to end all sessions[11].
3. **Find the way in.**
   Check for new admin accounts, recent plugins, changed PHP files and `.htaccess`[11]. Unless you fix the entry point, the site gets hacked again[10].
4. **Rebuild from clean sources.**
   Restore a backup from before the attack, or reinstall core and plugins from official copies[10,11]. Update everything, then change the passwords again[11].
5. **Tell the people who need to know.**
   If customer or card data may be affected, follow the incident response sheet and call your payment provider. Once clean, request a review in Google Search Console[12].

Updates, plugins and PHP

- **Check WordPress's settings.** Minor releases update automatically, but major ones only on sites installed since 5.6[13]. Enable both under Dashboard > Updates.
- **Buy premium plugins in the company's name.** WordPress shows updates for plugins installed from a ZIP file only if the vendor built in an updater[15]. Never use nulled copies from untrusted sources[3].
- **Replace abandoned plugins.** Check each plugin's last release once a quarter, and replace any that no longer get updates.
- **Patch PHP separately.** CMS updates don't fix the PHP runtime, and containers need a rebuild[18]. Set a PHP patch deadline with the hoster.

Online shops: scripts on the payment page

- **Know your setup.** PCI DSS 6.4.3 and 11.6.1 cover scripts on payment pages and on pages embedding the provider's iframe; with a redirect, confirm the scope with your acquirer or QSA[23]. SAQ A drops both, but iframe users must confirm their site resists script attacks[24].
- **Authorize every script (6.4.3).** Confirm each payment page script is authorized, assure its integrity and list it with a reason[23].
- **Detect changes (11.6.1).** Alert on unauthorized script and security header changes as browsers see them, weekly or per risk analysis[23].
- **Cut scripts from checkout.** Keep tag managers, chat widgets and pixels off payment pages, and limit sources with a Content Security Policy[23].

Admin access, agencies and hosters

- **One account per person.** No shared agency logins. Keep admin rights to a few named people and give content staff an editor role.
- **Prefer company SSO.** Sign admins in through your identity provider with SAML or OpenID Connect, so leavers lose access with their account.
- **Restrict the admin path.** Allow the backend only from VPN or office IPs, with TYPO3's `IPmaskList`[4] or web server rules, and limit failed logins[19].
- **Own every account.** Keep the domain, hosting, CDN, repositories and licenses in the company's name; registrar locks are in the DNS and domains sheet. When an agency leaves, remove its accounts and rotate SFTP, database and API credentials.

Hosting, backups, WAF and forms

- **Back up outside the hosting account.** Keep dated snapshots from before any compromise[3]. Restore tests are in the backups sheet.
- **Treat a WAF as time, not a fix.** For PHP's October flaws, the BSI said only patching or going offline protects fully[18]. Wordfence's free plan gets new firewall rules 30 days late[5].
- **Lock the origin behind your CDN.** Accept only CDN traffic, through an IP allowlist, authenticated origin pulls or a tunnel[25].
- **Restrict uploads and forms.** Fix allowed file types, sizes and folders, with no execute rights, and block bots without locking out real users[19].

End-of-support dates that set your schedule

- **PHP.** 8.1 and older get no releases; for October's flaws, the PHP project offered only source backports[20]. 8.2 ends on 31 Dec 2026, 8.3 on 31 Dec 2027[8].
- **Drupal.** Drupal 7 ended on 5 Jan 2025. Drupal 10 ends on 9 Dec 2026, the week 12.0 ships[9].
- **TYPO3.** Only v14 and v13 get free fixes[16]. v12 lost them on 30 Apr 2026; paid ELTS runs to 30 Apr 2029, or 2030 via a TYPO3 partner[21].
- **Joomla.** 5.x gets security fixes until 12 Oct 2027, 6.x until 16 Oct 2029[22].
- **WordPress.** Only the newest version is actively supported[14].
- **Shopware.** Its security plugin covers only versions still in support[17].

What changed in 2026

- **PHP needed an emergency patch.** On 21 Oct 2026, PHP fixed two critical unauthenticated code execution flaws[20]. CISA listed both as exploited on 26 Oct[26]. Check each site's PHP against its Linux distribution's patched package[18].
- **WordPress core entered CISA's KEV catalog.** Four core flaws were added from July to November 2026[26].
- **Hosting panels were exploited.** CISA lists a cPanel login bypass used in ransomware attacks, and a LiteSpeed cPanel plugin flaw that gave any cPanel user root[26].
- **Other platforms too.** CISA added a Drupal core flaw in May, flaws in five Joomla extensions in June and July, and three Adobe Commerce flaws since September[26].

What you can skip

- **Renaming the admin URL.** It doesn't stop an exploit. Restrict the path instead, and spend the time on updates and MFA.
- **A second or third security plugin.** Each one is more code to patch. One security plugin is enough.
- **Staging every security release.** WordPress strongly discourages turning off minor auto-updates[13]. Stage major upgrades only.
- **Perfect scores on header and TLS scanners.** Apply the encryption sheet's TLS baseline and a CSP on checkout, then get back to updates.

Mistakes that cost the most

- **Restoring a backup and calling it fixed.** Backdoors survive, such as a harmless-looking admin account or hidden PHP file[19].
- **Deleting only the obvious malware.** Attackers often inject code into existing files[19]. Replace core directories with clean copies[11].
- **Assuming someone else patches.** Check that the contract covers plugins, PHP and a fast response to a compromise[6].
- **Forgetting old sites and APIs.** Microsites, staging copies and shop APIs need care too. See the attack surface and API security sheets.

What this looks like at your size

Small team, no security staff

- Ask your agency or hoster in writing who updates core, plugins and PHP, and to fix exploited flaws within 72 hours.
- Turn on automatic updates and admin MFA.
- Take card payments on your provider's hosted payment page, not in your own form.

Growing, with IT or security staff

- List every site and staging copy with its owner, CMS, PHP version and end-of-support date.
- Sign admins in through your identity provider and allow the backend only from VPN or office IPs.
- Put a WAF or CDN in front, lock the origin to it and check files for unexpected changes[6].

Enterprise or regulated

- Track CMS, plugin and PHP advisories against the patching sheet's deadlines.
- Keep a checkout script inventory and change detection, with evidence for the assessor[23].
- Review web server and CMS logs regularly[6], and test shops after major releases.

Version history

5 Nov 2026, version 1.1	Merchants using a payment redirect now confirm the PCI script scope with their acquirer or QSA, since the PCI SSC supplement contradicts itself.

Sources

1. Plugin and themes auto-updates. WordPress.org, updated 13 Jan 2023. wordpress.org
2. State of WordPress Security in 2026 (vendor report). Patchstack, updated 25 Feb 2026. patchstack.com
3. Hardening WordPress. WordPress.org, updated 14 Oct 2026. developer.wordpress.org
4. TYPO3 Explained 14.3: BE backend configuration. TYPO3 docs, checked 5 Nov 2026. docs.typo3.org
5. Wordfence Security. WordPress.org, v9.0.2, 30 Sep 2026. wordpress.org
6. APP.3.2 Webserver. BSI IT-Grundschutz, Edition 2023. bsi.bund.de
7. § 19 TDDDG. BMJ, checked 5 Nov 2026. gesetze-im-internet.de
8. Supported Versions. PHP.net, checked 5 Nov 2026. php.net
9. Drupal core release schedule. Drupal.org, updated 3 Nov 2026. drupal.org
10. How to detect, analyze, and recover a hacked site. TYPO3 docs, rendered 4 Nov 2026. docs.typo3.org
11. FAQ My site was hacked. WordPress.org, updated 26 Jul 2026. wordpress.org
12. Security issues report. Google, checked 5 Nov 2026. support.google.com
13. Upgrading WordPress. WordPress.org, updated 28 Oct 2026. developer.wordpress.org
14. WordPress 7.1.4 (Security Release). WordPress.org, 28 Oct 2026. wordpress.org
15. Plugins screen. WordPress.org, updated 13 May 2026. wordpress.org
16. TYPO3 version support and security updates. TYPO3 docs, checked 5 Nov 2026. docs.typo3.org
17. Security Plugin. Shopware docs, checked 5 Nov 2026. developer.shopware.com
18. PHP: Kritische Schwachstellen bedrohen Webanwendungen weltweit. BSI, v1.2, 22 Oct 2026. bsi.bund.de
19. APP.3.1 Webanwendungen und Webservices. BSI IT-Grundschutz, Edition 2023. bsi.bund.de
20. Security advisory: Critical remote code execution vulnerabilities. PHP.net, 21 Oct 2026. php.net
21. Extended Long Term Support (ELTS). TYPO3 GmbH, checked 5 Nov 2026. typo3.com
22. Joomla! Project Roadmap. Joomla!, checked 5 Nov 2026. developer.joomla.org
23. Payment Page Security and Preventing E-Skimming (information supplement). PCI SSC, Mar 2025. pcisecuritystandards.org
24. FAQ 1588: SAQ A eligibility criteria for scripts. PCI SSC, Feb 2025. pcisecuritystandards.org
25. Protect your origin server. Cloudflare, updated 2 Oct 2026. developers.cloudflare.com
26. Known Exploited Vulnerabilities Catalog. CISA, version 2026.11.04. cisa.gov