Plugin Security Certification (PSC) by CleanTalk

Use only certified WordPress plugins for your website

CERTIFY PLUGIN

CVE-2025-10583 – WP Fastest Cache – Missing Authorization to Authenticated (Subscriber+) Blind Server-Side Request Forgery – POC

CVE-2025-10583 is an **authenticated Server-Side Request Forgery (SSRF)** vulnerability in the WordPress plugin **WP Fastest Cache**, affecting versions **up to and including 1.7.4** according to the NVD record. What makes this issue especially operationally relevant is the plugin's adoption: the WordPress.org listing shows **1+ million active installations**, so any low-privilege-to-network-recon bug has immediate "real internet" consequences across a large attack surface. The core impact is not a direct data exfiltration primitive by itself, but rather a reliable way for a low-privileged authenticated user to coerce the server into making outbound connections, which can be weaponized for internal network discovery, firewall bypass, and chaining into higher-impact compromises.

CVE	**CVE-2025-10583**
Plugin Version	WP Fastest Cache Premium <= 1.7.4

All Time	**62 858 825**
Active installations	**1 000 000+**
Publicly Published	December 11, 2025
Last Updated	December 11, 2025
Researcher	Dmitrii Ignatyev
PoC	Yes
Exploit	No
Reference	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2025-10583 https://www.wordfence.com/threat-intel/vulnerabilities/wordpress-plugins/wp-fastest-cache-premium/wp-fastest-cache-premium-174-missing-authorization-to-authenticated-subscriber-blind-server-side-request-forgery https://t.me/cleantalk_researches/363
Plugin Security Certification by CleanTalk	

Logo of the plugin	

Timeline

August 28, 2025	Plugin testing and vulnerability detection in the **WP Fastest Cache** have been completed
August 28, 2025	I contacted the author of the plugin and provided a vulnerability PoC with a description and recommendations for fixing
December 11, 2025	Registered **CVE-2025-10583**

Discovery of the Vulnerability

The Description attributes the issue to the `get_server_time_ajax_request` AJAX action, which can be abused by authenticated attackers with **Subscriber-level access and above** to make requests to arbitrary locations

originating from the server. In your technical analysis, the vulnerable implementation pattern is consistent with “missing authorization + missing nonce”: the premium module registers the AJAX action without enforcing an appropriate `current_user_can()`gate and without requiring a nonce, then reads attacker-controlled URLs from `servers[i][url]` and initiates a network connection using `fsockopen()` with a short timeout. That combination matters because it turns what is presumably a “latency/time check” feature into a **blind connection oracle**: the attacker may not see full response content, but they can still learn whether a host:port is reachable based on timing and success/failure behavior.

Understanding of SSRF attack’s

Even though SSRF is not “sensitive data exposure” in the classic sense (like leaking `wp-config.php`), in WordPress environments it frequently becomes the **bridge** to sensitive information because the WordPress server often has network reachability that external attackers do not. Common examples include metadata services (cloud instance metadata endpoints), internal databases, internal admin panels, monitoring dashboards, Redis/Memcached, search backends, or “localhost-only” management interfaces that rely on network position for security rather than strong authentication. With SSRF, the attacker stops being “outside the perimeter” and starts leveraging the server as a proxy into otherwise unreachable network segments; this is why the NVD notes it can be used to “query and modify information from internal services,” even when the initial primitive looks like a simple outbound request mechanism. In practice, the most immediate harm is often **reconnaissance** (mapping internal services), but the downstream risk is **credential theft, configuration disclosure, or lateral movement** if any internal service is misconfigured or exposes sensitive operations to trusted subnets.

Exploiting the SSRF Vulnerability

To exploit **CVE-2025-10583**, an attacker with Subscriber+ cookies:

> ***POC:***
>
> ```
> http://127.0.0.1/wordpress/wp-admin/admin-ajax.php?action=get_server_time_ajax_request&servers[0][url]=http://127.0.0.1:444
> ```
>
> ———

The practical risk profile of CVE-2025-10583 is **internal network reconnaissance and pivot enablement from a low-privilege account**, not a single-click RCE. Once an attacker has a Subscriber account (which is common on membership sites, stores, LMS platforms, and community portals), they can use the WordPress server as a vantage point to probe internal RFC1918 address space, localhost-bound services, or cloud metadata endpoints that are frequently blocked from the public internet but reachable from the instance itself. This can directly undermine segmentation assumptions (“the admin panel is safe because it’s only on the internal network”), and it can expose hidden management services that were never intended to face untrusted users. Even when the SSRF is “blind,” it can still materially increase attack success by identifying where high-value services live, which ports are open, and which targets respond differently (fast connect vs timeout), enabling attackers to focus subsequent exploitation attempts where it’s most likely to work.

Recommendations for Improved Security

The remediation is conceptually simple: treat this endpoint as a privileged diagnostic action and enforce **authorization, request authenticity, and strict destination validation**. At minimum, the handler should require a strong capability gate (e.g., `manage_options` or a plugin-specific admin capability)

and must require a nonce to prevent CSRF, because state-changing or network-active AJAX actions should not be callable by arbitrary authenticated users. The NVD's weakness classification aligns with **Missing Authorization**, reinforcing that the primary fix is a correct capability check rather than superficial sanitization. On top of that, harden the network behavior: apply an allow-list of permitted targets (or disallow private/loopback/link-local ranges entirely), restrict ports, enforce a safe URL parser, and ensure the endpoint does not act as a timing oracle (for example, by returning a consistent response without leaking connect/timeout differences). Operationally, site owners should patch/upgrade beyond the vulnerable range (NVD: up to 1.7.4), review whether premium diagnostic features are exposed to non-admin roles, and consider egress controls at the infrastructure layer so that even if an SSRF primitive exists, it cannot reach sensitive internal destinations.

> *By taking proactive measures to address **SSRF vulnerabilities like CVE-2025-13922** WordPress website owners can enhance their security posture and safeguard against potential exploitation. Stay vigilant, stay secure.*
>
> *#WordPressSecurity #**SSRF** #WebsiteSafety #StayProtected #HighVulnerability*
>
> ***Use CleanTalk solutions to improve the security of your website***
>
> DMITRII I.

Related posts

CVE-2024-0756 – Insert or Embed Articulate Content into WordPress – Stored XSS/ Iframe Injection – POC

WordPress, a leading content management system, is widely used for creating websites due to its flexibility and extensive...

CVE-2024-7315 – Migration, Backup, Staging – WPvivid – Unauth Sensitive Data Exposure and Database password leak – POC

A critical vulnerability, designated as CVE-2024-7315, has been discovered in the WPvivid plugin, widely used for migration, backup,...

CVE-2024-4004 – Advanced Cron Manager – Stored XSS to JS backdoor – POC

CVE-2024-4004 is a newly discovered Stored Cross-Site Scripting (XSS) vulnerability in the widely used WordPress plugin Advanced Cron...

CVE-2024-10637 – Kadence Blocks – Stored XSS to JS Backdoor Creation – POC

Kadence Blocks, a popular WordPress plugin used to extend the functionality of the Kadence theme by adding custom...

CVE-2024-10102 – Robo Gallery (Photo Gallery, Images, Slider in Rbs Image Gallery) – Stored XSS to JS Backdoor Creation – POC

Robo Gallery, a popular WordPress plugin used for displaying photo galleries and sliders, contains a critical vulnerability, CVE-2024-10102....

Dmitrii I February 4, 2026 CVE, Security

No Comments

← Plugin Security Certification (PSC-2026-64603): “Google for WooCommerce” – Version 3.5.2

Plugin Security Certification (PSC-2026-64604): “Wordfence Security” – Version 8.1.4 →

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment *

Name *

Email *

Website

Save my name, email, and website in this browser for the next time I comment.

Post Comment

CleanTalk

- About
- Blog
- Dashboard
- Plugins security certification
- WordPress Malware removal
- Pricing
- Sign up / Sign In

Solutions

- Anti-Spam for Websites
- Anti-Spam Plugins
- Check IP
- Check Email
- Filter fake emails
- Gantt Charts
- Hide contact data
- Stop spam emails in Contact form 7 (CF7)
- Stop spam in Elementor form builder
- Stop spam in WPForms
- Vulnerabilities and Security Researches
- Website malware scanner
- WordPress Security & Firewall Plugin

Documentation

- Help
- Is my site infected?
- License agreement
- Privacy Policy
- Refund Policy

Contact us

- Create a support ticket
- Contact us
- Telegram
- X