

HiveForce Labs

THREAT ADVISORY

 VULNERABILITY REPORT

Race Against the Patch: Critical WordPress RCE Exploited Within Hours

Date of Publication

September 25, 2026

Admiralty Code

A1

TA Number

TA2026283

Summary

First Seen: September 22, 2026

Affected Products: WordPress (WordPress Core)

Impact: CVE-2026-87902 is a critical, unauthenticated flaw in WordPress core, rated CVSS 9.2 (CWE-98, PHP file inclusion). By sending a specially encoded path in a page request, an attacker can trick the `get_page_template()` function into loading a PHP file from outside the theme folder, and where the server and active theme meet a few common conditions, that becomes full remote code execution with no login required. It affects a wide range of versions, from 4.7.0 to 7.1.1. There is no named group behind it: attackers reverse-engineered the fix as soon as WordPress shipped version 7.1.2 and started scanning the same day, quickly moving from simple checks to writing malicious PHP onto vulnerable servers. With a public exploit tool already circulating, any site that meets the preconditions should be treated as at risk of full compromise.

CVE

CVE	NAME	AFFECTED PRODUCT	ZER O-DAY	CISA KEV	PATC H
CVE-2026-87902	WordPress Unauthenticated Path Traversal Vulnerability	WordPress WordPress Core			

Vulnerability Details

#1

At its core, this is a PHP file inclusion flaw (CWE-98) hiding inside WordPress's page-template logic. Attackers wasted no time with it: within hours of WordPress going public with the vulnerability, live sites were already being probed and hit. When WordPress builds a page, `get_page_template()` decides which template file to load based on the page's slug. WordPress runs that slug through its own sanitiser first, and that sanitiser was written to preserve percent-encoded characters while stripping literal dots and slashes. That gap is the whole problem, path is blocked, but a percent-encoded version survives the sanitiser and is then decoded, letting the template path climb out of the theme directory and point at any readable .php file on the server.

#2

To pull this off, an attacker sends a request that pairs a valid `page_id` with a crafted `pagename` value. The `page_id` matters, because without a real page WordPress returns a 404 and the vulnerable code never runs, which is also why the two parameters appearing together is such a useful detection signal. The `pagename` value usually begins with a real directory starting with `page-` (such as `page-templates`), then uses the encoded traversal to reach a target file. The favourite target is PHP's `pearcmd.php`.

#3

When the server has `register_argc_argv` enabled, the query string is handed to the included script as command-line arguments. An attacker first sends `+config-show` to confirm the trick works, then switches to `+config-create` to write a file of their choosing, arbitrary PHP with attacker-controlled content, which is code execution. Observed activity drops these files into `/tmp` and `/var/tmp`; some are harmless proof-of-concept markers, while others plant a tag that runs a shell command or pulls in a web shell.

#4

The vulnerability affects WordPress core versions 4.7.0 through 7.1.1, an unusually long range that reaches back years. The "attack requirements present" element reflects the two preconditions: the active theme must have a top-level directory whose name starts with `page-`, and a suitable readable .php target (such as `pearcmd.php`) must exist on the server. This is not theoretical, the first exploitation attempts hit sensors at 11:49 UTC on 22 September 2026, the very day the patch shipped, showing the payloads were built straight from the patch diff. By that afternoon the activity had moved from probing to writing files to disk, and by 23 September a named Nuclei template was circulating and traffic had spiked, with independent honeypot telemetry recording dozens of attempts inside the first day.

Vulnerability

CVE ID	AFFECTED PRODUCTS	AFFECTED CPE	CWE ID
CVE-2026-87902	WordPress WordPress Core (4.7.0 – 7.1.1)	cpe:2.3:a:wordpress:wordpress:*:*:*:*:*	CWE-98

Recommendations



Update WordPress Immediately: Apply WordPress 7.1.2 right away, or the patched release for your branch, 7.0.6, 6.9.9, 6.8.10, and backports reaching all the way to 4.7.37. This is the only complete fix, and with active exploitation under way it should be treated as urgent rather than as routine maintenance.



Hunt for Signs of Compromise: Check `/tmp` and `/var/tmp` for unexpected `.php` files, including names like `wp-pear-rce-flag.php`, `poc87902.php`, and files beginning with `luci_` or `zeta_`. Search web logs for a `pagename` value containing `%2e%2e` or `%252e%252e`, a `pagename` beginning with `templates%2f`, `pagename` and `page_id` appearing together, or any request mentioning `pearcmd`, `+config-show`, or `+config-create`. If a normal page URL ever returned OPML or RSS content, treat that host as successfully probed and investigate it as potentially compromised.



Apply Temporary Mitigations If You Cannot Patch Yet: If an immediate update is not possible, block requests where the `pagename` parameter contains traversal sequences; a genuine page slug never does, so this can be filtered without affecting normal traffic. Disabling PHP's `register_argc_argv` setting does not fix the inclusion itself, but it breaks the `pearcmd` chain, downgrading potential code execution to a lower-impact information leak.



Harden the PHP Environment: Remove or restrict PEAR and `pearcmd.php` on production web servers where they are not needed, since they are the primary springboard from file inclusion to full code execution. Run the web server under a least-privilege account so that even a successful inclusion has limited reach.



Vulnerability Management: Maintain an accurate inventory of your WordPress installations and their versions, keep automatic updates enabled, and place a virtual patching or web application firewall layer in front of internet-facing sites so newly disclosed flaws can be blocked quickly while patches roll out. Review the security posture of third-party themes and plugins regularly as part of this process.



Potential MITRE ATT&CK TTPs

Tactic	Technique	Sub-technique
Reconnaissance	<u>T1595</u> : Active Scanning	<u>T1595.002</u> : Vulnerability Scanning
Initial Access	<u>T1190</u> : Exploit Public-Facing Application	
Execution	<u>T1059</u> : Command and Scripting Interpreter	
Resource Development	<u>T1588</u> : Obtain Capabilities	<u>T1588.006</u> : Vulnerabilities



Indicators of Compromise (IOCs)

TYPE	VALUE
IPv4	43[.]250[.]53[.]42, 180[.]251[.]159[.]243, 195[.]178[.]110[.]247, 107[.]189[.]14[.]87, 45[.]61[.]184[.]170, 92[.]246[.]130[.]76
Filename	wp-pear-rce-flag.php, poc87902.php, luci_<random>.php, zeta_<random>.php
File Path	/usr/local/lib/php/pearcmd.php, /usr/share/php/pearcmd.php, /usr/share/pear/pearcmd.php



Patch Link

<https://wordpress.org/download/releases/>



References

<https://github.com/WordPress/wordpress-develop/security/advisories/GHSA-7hp8-65ch-5whp>

<https://patchstack.com/articles/cve-2026-87902-attackers-started-probing-wordpress-sites-hours-after-the-patch/>



What Next?

At Hive Pro, it is our mission to detect the most likely threats to your organization and to help you prevent them from happening.

Book a Demo of HivePro.

REPORT GENERATED ON

September 25, 2026 • 07:20 AM

© 2026 All Rights are Reserved by Hive Pro



More at www.hivepro.com