Hive Pro

Threat Level

Red

Hiveforce Labs

THREAT ADVISORY

VULNERABILITY REPORT

CVE-2026-64638 (XSS2Shell): Pre-Auth XSS in the WordPress Login Screen

Date of Publication	Admiralty Code	TA Number
August 10, 2026	A1	TA2026227

Summary

First Seen: July 26, 2026
Affected Product: WordPress
Impact: WordPress has patched CVE-2026-64638, nicknamed XSS2Shell, a pre-authentication reflected cross-site scripting flaw in the Core login screen. The immediate consequence of CVE-2026-64638 is unauthenticated JavaScript execution in the trusted origin of an affected WordPress site, which permits credential harvesting presented on a legitimate hostname, theft of session material, and arbitrary same-origin requests issued in the victim's security context.

CVE

CVE	NAME	AFFECTED PRODUCT	ZERO-DAY	CISA KEV	PATCH
CVE-2026-64638	XSS2Shell (WordPress Core Pre-Auth Reflected Cross-Site Scripting Vulnerability)	WordPress Core	✗	✗	✓

Vulnerability Details

#1

WordPress has remediated the CVE-2026-64638 flaw, named XSS2Shell, a pre-authentication reflected cross-site scripting vulnerability in the WordPress Core login interface. The flaw resides in the handling of the username value submitted in a failed login attempt at wp-login.php.

#2

What distinguishes CVE-2026-64638 from a routine reflected XSS finding is a publicly documented escalation path that terminates in server-side PHP execution. The root cause is a parser disagreement between two sanitization routines applied in sequence to the same value: attacker-controlled markup survives the first routine as inert text and is then promoted to live HTML elements by the second.

#3 Exploitation requires no account, no session, and no prior knowledge of the target. A single failed login attempt is sufficient to place attacker-chosen DOM elements on the rendered error page and drive WordPress's own bundled JavaScript into script execution within the site's origin.

#4 Against a victim already authenticated as a single-site Administrator who interacts with an attacker-controlled page, a full escalation chain that abuses the native Application Password approval flow, the REST API, and the plugin upload facility to achieve PHP code execution as the web server user.

#5 No threat actor has been attributed to this vulnerability and no malware has been associated with it. Because WordPress Core powers a very large share of internet-facing websites and the affected login interface is by design publicly reachable, the exposed population is substantial, even though escalation to server-side code execution depends on several additional preconditions aligning.

Vulnerability

CVE ID	AFFECTED PRODUCTS	AFFECTED CPE	CWE ID
CVE-2026-64638	WordPress Core 4.7.0 through 7.0.2, all branches; before 7.0.3	cpe:2.3:a:wordpress:wordpress:*:*:*:*:*:*:*:*	CWE-79

Recommendations

Update WordPress Core Immediately: Apply WordPress 7.0.3, or the equivalent patched minor release for your supported branch such as 6.9.6, 6.8.7, or 6.7.6, without delay. This is the only fix for the underlying flaw, and every other measure described below reduces blast radius without closing the vulnerability. Verify the running version directly through the dashboard, hosting control panel, WP-CLI, or asset inventory rather than assuming that a configured automatic update policy has applied; sites relying on automatic background updates should have received the release, but this must be confirmed rather than trusted.

Inventory Every WordPress Instance and Record Its Core Version: Build or refresh a complete inventory of WordPress installations across the estate, including instances that fall outside centralized patch management. Departmental, laboratory, research group, campaign, and course sites are the highest-risk population precisely because they are internet-facing and self-administered, and these are frequently absent from the asset register that the security team actually works from. Record the Core version for each so that remediation can be tracked to completion rather than assumed.

Migrate or Decommission Unsupported Installations: Installations running WordPress 4.6 or earlier are end of life and will receive no patch for this vulnerability. Any such instance must be migrated onto a supported and patched branch, isolated from public reachability, or decommissioned outright. Treat continued public exposure of an unsupported Core version as an accepted risk requiring explicit sign-off rather than an operational detail.

Enforce DISALLOW_FILE_MODS and Restrict PHP Execution From Plugin Directories: Defining DISALLOW_FILE_MODS in wp-config.php breaks the plugin upload stage of the chain and is sound configuration in its own right on any installation where the plugin lifecycle is managed by deployment rather than through the administrative interface. Where feasible, additionally ensure that hardening prevents direct PHP execution from inactive plugin directories, since the demonstrated chain reaches code execution without ever activating the uploaded plugin. As with the preceding recommendation, these constrain the escalation path only.

Reduce the Administrator Population: Accounts whose function is limited to content publication should hold the Editor role rather than Administrator. The escalation chain depends specifically on the default unfiltered_html and upload_plugins capabilities held by single-site Administrators, so narrowing that population directly narrows the set of users whose compromised browser session can be converted into server-side code execution.

Potential MITRE ATT&CK TTPs

Tactic	Technique	Sub-technique
Initial Access	T1190: Exploit Public-Facing Application	
	T1189: Drive-by Compromise	
Execution	T1059: Command and Scripting Interpreter	T1059.007: JavaScript
Defense Evasion	T1211: Exploitation for Defense Evasion	
Credential Access	T1528: Steal Application Access Token	
	T1552: Unsecured Credentials	T1552.001: Credentials In Files
Persistence	T1505: Server Software Component	T1505.003: Web Shell
	T1136: Create Account	
Privilege Escalation	T1548: Abuse Elevation Control Mechanism	

Patch Link

https://wordpress.org/news/2026/08/wordpress-7-0-3-release/

References

https://github.com/WordPress/wordpress-develop/security/advisories/GHSA-52p2-r8wf-jcrf

https://pwn.ai/blog/xss2shell

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

August 10, 2026 • 04:00 AM

More at www.hivepro.com