Hive Pro

Threat Level

Red

Hiveforce Labs

THREAT ADVISORY

VULNERABILITY REPORT

wp2shell: One Request, No Login, Full Control of WordPress Core

Date of Publication	Admiralty Code	TA Number
July 20, 2026	A1	TA2026205

Summary

First Seen: July 2026
Affected Products: WordPress Core 6.8.x, 6.9.x, and 7.0.x (7.1 beta also affected)
Impact: WordPress Core, which powers more than 500 million websites, has a serious flaw chain called wp2shell that lets a remote attacker take over a site with a single web request, with no login or plugins needed. It combines two bugs: CVE-2026-63030, a REST API batch-route confusion introduced in WordPress 6.9, and CVE-2026-60137, an SQL injection in the author__not_in parameter of WP_Query. On WordPress 6.9 and 7.0, the two chain together into full remote code execution. WordPress released fixes (7.0.2, 6.9.5, 6.8.6, and 7.1 beta2) on July 17, 2026, and switched on forced auto-updates, but working exploits are already circulating on GitHub, so any site not yet confirmed on a patched version should be updated immediately.

CVEs

CVE	NAME	AFFECTED PRODUCT	ZERO-DAY	CISA KEV	PATCH
CVE-2026-63030	wp2shell (WordPress Remote Code Execution Vulnerability)	WordPress WordPress Core	✗	✗	✓
CVE-2026-60137	wp2shell (WordPress SQL Injection Vulnerability)	WordPress WordPress Core	✗	✗	✓

Vulnerability Details

#1

wp2shell is not a single vulnerability but a chain of two distinct weaknesses in WordPress core that, when combined, allow an anonymous attacker to move from an ordinary web request to remote code execution. The researchers describe an attack with no preconditions: it needs no login, no plugins, and no special settings, only a reachable WordPress site running an affected version. That combination of zero prerequisites and a target base measured in the hundreds of millions of sites is what makes the chain so significant.

#2

The first flaw in the chain is CVE-2026-63030, a REST API batch-route confusion weakness that WordPress rates as critical. The batch endpoint itself is a long-standing feature that has shipped since WordPress 5.6 and lets clients bundle several REST calls into one request, but a change introduced in the 6.9 release created a route-handling confusion that an unauthenticated caller can abuse.

#3

The second flaw is CVE-2026-60137, an SQL injection flaw in the author__not_in parameter of the WP_Query class. WordPress does not properly sanitise this parameter, so untrusted input that reaches it can be used to inject arbitrary SQL. Analysis maps this to CWE-89 (SQL Injection) and scores it CVSS 3.1 9.1, driven by the potential to both read and modify database contents over the network without authentication. The vendor's own advisory labels this issue as moderate in isolation, which reflects that, on its own, it depends on how a given site exposes the parameter.

#4

The danger comes from how the two flaws interlock. On WordPress 6.9 and later, the batch-route confusion gives an anonymous request a way to reach code paths that feed the vulnerable query logic, turning the SQL injection into a remotely triggerable primitive that ultimately yields code execution. On WordPress 6.8 the SQL injection is present but cannot be chained this way, because the batch-route confusion bug did not exist before 6.9, so 6.8 sites face data exposure rather than full compromise. Notably, the underlying batch framework and its request format have been publicly documented for years, and the published material so far does not explain exactly what changed in 6.9 to open the door.

#5

In terms of scope, the complete remote code execution chain affects WordPress 6.9.0 through 6.9.4 and 7.0.0 through 7.0.1, while the SQL injection issue on its own reaches back to 6.8.0 through 6.8.5. Fixes landed in WordPress 7.0.2 and 6.9.5, with 6.8.6 addressing the SQL injection on the older branch and 7.1 beta2 covering the pre-release line. The 7.0.2 update touched three core files tied to REST request handling and query construction, and WordPress enabled forced background updates so that many affected sites would be patched automatically.

#6

Multiple proof-of-concept exploits were posted to GitHub. Some of them abuse SQL injection to extract WordPress password hashes, crack an administrator credential offline, then log in, upload a malicious plugin, and run commands, while others claim true pre-authentication code execution without any credentials, matching the researchers' original description.

Vulnerabilities

CVE ID	AFFECTED PRODUCTS	AFFECTED CPE	CWE ID
CVE-2026-63030	WordPress WordPress Core (Version before 6.9.5 and 7.0.2)	cpe:2.3:a:wordpress:wordpress:*:*:*:*:*:*:*:*	CWE-436
CVE-2026-60137	WordPress WordPress Core (Version before 6.9.5 and 7.0.2)	cpe:2.3:a:wordpress:wordpress:*:*:*:*:*:*:*:*	CWE-89

Recommendations

Update WordPress Immediately: Move every affected site to WordPress 7.0.2 or 6.9.5 without delay, and update 6.8 sites to 6.8.6 to close the SQL injection issue on that branch. This is the only complete fix for the wp2shell chain, and with public exploits already circulating, patching is the single most important action you can take right now.

Confirm the Update Actually Landed: WordPress enabled forced auto-updates, but do not assume the push reached every site, especially those with automatic updates disabled. Check the actual version each site is running from the dashboard rather than trusting that the update applied, and manually update any instance still on a vulnerable release.

Apply Temporary Restrictions If You Cannot Patch Yet: If updating is not immediately possible, block anonymous access to the batch endpoint as an emergency stopgap. At the WAF level this means blocking both /wp-json/batch/v1 and the ?rest_route=/batch/v1 query-string route, since a rule covering only the first path leaves the second exposed. Alternatively, restrict or disable unauthenticated REST API access with a suitable plugin. Treat these as temporary measures only, as they can break legitimate integrations, and remove them once you have patched.

Rotate Credentials and Review Accounts: On any site that may have been exposed, reset administrator and database passwords and force a re-authentication, since the SQL injection can expose stored password hashes. Remove any account you cannot account for and confirm that the remaining privileged users are legitimate.

Potential MITRE ATT&CK TTPs

Tactic	Technique	Sub-technique
Initial Access	T1190: Exploit Public-Facing Application	
Execution	T1059: Command and Scripting Interpreter	
Credential Access	T1110: Brute Force	T1110.002: Password Cracking
Resource Development	T1588: Obtain Capabilities	T1588.006: Vulnerabilities
Privilege Escalation	T1078: Valid Accounts	
Persistence	T1505: Server Software Component	

Patch Link

https://wordpress.org/news/2026/07/wordpress-7-0-2-release/

References

https://slcyber.io/research-center/wp2shell-pre-authentication-rce-in-wordpress-core/

https://socradar.io/blog/wp2shell-wordpress-rce-cve-2026-63030/

https://github.com/WordPress/wordpress-develop/security/advisories/GHSA-ff9f-jf42-662q

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

July 20, 2026 • 07:30 AM

© 2026 All Rights are Reserved by Hive Pro

More at www.hivepro.com