NCA-14.230726 – National CERT Advisory

Critical Unauthenticated Remote Code Execution Vulnerabilities in WordPress Core

An elevated cyber risk posture has been assessed following an emerging threat and a rising volume of attacks targeting critical vulnerabilities in WordPress Core.

These vulnerabilities allow unauthenticated remote attackers to execute arbitrary commands and achieve full website takeover with no prior login or user interaction required. The primary vulnerability, publicly identified as **wp2shell** (CVE-2026-63030), is a REST API batch-route confusion flaw at the /wp-json/batch/v1 endpoint that chains with a core SQL injection vulnerability (CVE-2026-60137) in the WP_Query class.

Proof-of-concept material is actively circulating, and exploitation activity was observed within hours of disclosure. Any public-facing website built on WordPress should be treated as at high risk until updated. Web and IT teams are strongly advised to apply the latest WordPress Core security updates as an immediate priority and confirm that every site is current.

DOCUMENT SUMMARY

Field	Details
Advisory ID	NCA-14.230726
Threat Type	Unauthenticated Remote Code Execution (RCE), SQL Injection, Web Shell Deployment, Server-Side Foothold
Severity	Critical (Active Exploitation Observed / PoC Public)
Attack Vector	Remote network exploitation via REST API (/wp-json/batch/v1 endpoint)
Authentication Required	None
User Interaction	Not Required
CVSS Score	9.8 (Critical – CVE-2026-63030) / 9.1 (Critical – CVE-2026-60137)
Scope & Applicability	Government portals, CII, enterprise websites, financial institutions, and public hosting environments using WordPress
Affected Products	WordPress Core versions 6.9.0 through 7.0.1 (RCE chain), 6.8.x and later (SQLi)
Security Advisory Source	WordPress Core Release Notes & Security Advisories

IMPACT ANALYSIS

Failure to remediate these vulnerabilities may result in:

1. Unauthenticated remote code execution on web servers
2. Complete website compromise and administrative takeover
3. Unauthorized SQL injection and sensitive data exposure
4. Deployment of persistent web shells in writable directories
5. Establishment of a server-side foothold for lateral network movement

6. Defacement or unauthorized modification of official public portals
7. Cross-site contamination in multi-tenant or shared web hosting environments
8. Disruption of web services and digital public services
9. Potential compromise of interconnected enterprise databases
10. Reputational damage and loss of trust in public digital infrastructure

THREAT CHARACTERISTICS

Characteristic	Details
Threat Category	Content Management System (CMS) Core Vulnerability
Threat Status	Active exploitation attempts & public PoC observed
Root Cause	REST API batch-route confusion chained with WP_Query SQL Injection
Attack Surface	Internet-facing WordPress REST API (/wp-json/batch/v1 or rest_route=/batch/v1)
Attack Complexity	Low (remote unauthenticated exploitation feasible)
Privileges Required	None
Impact Scope	Full website and underlying server compromise
CWE Classes	CWE-94 (Code Injection), CWE-89 (SQL Injection), CWE-863 (Incorrect Authorization)

VULNERABILITY DETAILS

- **CVE-2026-63030 (wp2shell):** Critical REST API batch-route confusion issue (/wp-json/batch/v1) that chains with a core SQL injection flaw to achieve unauthenticated remote code execution.
  - *RCE Chain Affected Versions:* 7.0.0 through 7.0.1 (Fixed in **7.0.2**); 6.9.0 through 6.9.4 (Fixed in **6.9.5**).
- **CVE-2026-60137:** Core SQL injection vulnerability in the WP_Query class.
  - *SQLi Affected Versions:* 6.8.x and later (Fixed in **6.8.6**, **6.9.5**, and **7.0.2**).

INDICATORS OF EXPOSURE / TARGETING (IoEs)

Organizations should actively hunt for the following indicators within web server access logs, WAF telemetry, and file system audit logs:

1. **Suspicious Activity Patterns**
   - High-frequency automated POST/GET requests targeting /wp-json/batch/v1 or query parameter rest_route=/batch/v1.
   - Unauthenticated REST batch requests containing SQL injection payloads targeting WP_Query parameters.
2. **Exploitation Behavior Indicators**
   - Creation or modification of unauthorized .php files in writable directories (e.g., /wp-content/uploads/).
   - Web server processes (e.g., www-data, apache, nginx) spawning system shells or execution utilities.
   - Unexpected administrative account creation or privilege escalations in WordPress database tables.

3. **Network Indicators**
    - Outbound command-and-control (C2) connections originating from the web server to unclassified external IP addresses.
    - Automated vulnerability scanning traffic targeting REST API endpoints.

REMEDIATION ACTIONS

1. Immediate Defensive Controls

Action Category	Recommended Actions
Patch Management	Ask web or IT teams to update WordPress Core immediately to **7.0.2**, **6.9.5**, or **6.8.6** depending on the running release branch.
Version Verification	Manually verify the running version on every internet-facing site; do not rely solely on automatic updates as managed environments may block them.
WAF Mitigation	If immediate patching is not feasible, block requests at the WAF/web server to path /wp-json/batch/v1 and parameter rest_route=/batch/v1.
Access Restriction	Restrict unauthenticated access to the REST API batch routes via security plugins or custom must-use plugins.
Component Hardening	Update all plugins, themes, and must-use plugins in the same maintenance window.
Environment Isolation	Place staging or internal sites behind an authenticated or IP-restricted network boundary.
Temporary Endpoint Blocking	**If patching is delayed, block anonymous access to /wp-json/batch/v1 and rest_route=/batch/v1 at the WAF or web-server level.**
Core File Integrity Check	Validate WordPress core files against official release checksums to detect tampering or unauthorized modification.
Unauthorized File Hunt	Inspect writable directories, especially **/wp-content/uploads/**, for newly created or modified **.php** files and possible web shells.
Log Correlation	Correlate WAF logs, web server access logs, and file-integrity monitoring data to identify exploitation patterns.
Credential and Secret Rotation	Rotate WordPress administrative credentials and relevant database secrets after patching or any suspected exposure.
Plugin and Theme Review	Verify that all plugins, themes, and must-use plugins are fully updated and remove unnecessary or unused components.
API Exception Review	Where REST API access is operationally required, document approved use cases and restrict access through authentication or IP allowlists.

2. Enhanced Monitoring & Detection

- Enable detailed web server access logging and audit trails.
- Deploy SIEM detection rules for exploitation attempts directed at REST batch endpoints.
- Monitor for unexpected process execution spawned by web service contexts.
- Perform file-integrity monitoring to detect new PHP files in writable directories.

3. Incident Response Requirements

Organizations should:

- Audit all public-facing WordPress instances immediately.
- Investigate access logs for incoming requests to /wp-json/batch/v1.
- Isolate potentially compromised web servers immediately and preserve logs for forensic analysis.
- Conduct full administrative credential and database secret rotations following patching or suspected exposure.

ACTION SUMMARY & RESPONSE PRIORITIES

Action Type	Specific Steps
Patch Deployment	Apply WordPress Core security updates (7.0.2 / 6.9.5 / 6.8.6)
Deployment Verification	Manually verify running version across every public site
Exposure Reduction	Block /wp-json/batch/v1 at WAF if patching is delayed
Threat Hunting	Search web logs for REST API batch exploitation patterns
File Audit	Inspect writable directories for unauthorized PHP web shells
Incident Escalation	Report confirmed compromises to National CERT

MONITORING & DETECTION REQUIREMENTS

1. Monitor unauthorized access attempts to /wp-json/batch/v1.
2. Track anomalous REST API request structures and parameters.
3. Identify process spawning anomalies originating from web server services.
4. Monitor writable file system directories for new .php script creation.
5. Track database queries attempting SQL injection against WP_Query.
6. Detect external scanning targeting WordPress REST endpoints.
7. Validate core file integrity against official release checksums.
8. Correlate WAF block logs with web server execution telemetry.
9. Monitor lateral movement attempts from web server hosts into internal networks.
10. Track anomalous outbound connections from web hosting infrastructure.

INCIDENT REPORTING

All suspected compromises, exploitation attempts, or anomalous activity must be immediately reported to National CERT Pakistan:

- **Incident Reporting Portal:** https://pkcert.gov.pk/report-incident/
- **Email:** cert@pkcert.gov.pk
- **UAN:** +92 519203412

CALL TO ACTION – KEY IMPERATIVES

#	Requirement
1	Apply WordPress Core security patches immediately.
2	Confirm every public-facing site is running a fixed version.
3	Implement WAF mitigation rules if patching cannot be applied instantly.
4	Audit web server directories for unauthorized files or backdoors.
5	Update all accompanying plugins, themes, and extensions.
6	Restrict unauthenticated access to REST API batch endpoints.
7	Escalate confirmed incidents or web shell discoveries to National CERT.
8	Maintain continuous log monitoring across all web environments.