GOVERNMENT OF THE REPUBLIC OF VANUATU

PRIME MINISTER'S OFFICE

CERTVU
DEPARTMENT OF COMMUNICATIONS & DIGITAL TRANSFORMATION

PM B 9108 Port Vila, Vanuatu
Tel: (678) 33380

GOUVERNEMENT DE LA REPUBLIQUE DU VANUATU

BUREAU DU PREMIER MINISTRE

CERTVU

DEPARTMENT DE COMMUNICATION ET DE TRANSFORMATION NUMERIQUE

SPP 9108 Port Vila, Vanuatu
Tel: (678) 33380

10 July 2026

Advisory 163: Large-Scale CMS Exploitation Campaign (Webshell Deployment)

Release Date: 09th July 2026
Impact: HIGH / CRITICAL
TLP: CLEAR

The Department of Communications and Digital Transformation (DCDT) through CERT Vanuatu (CERTVU), provides the following advisory.

This alert is relevant to Organizations and System/Network administrators that utilize the above products. This alert is intended to be understood by technical users and systems administrators.

What is it?

A large-scale, globally active exploitation campaign is targeting multiple vulnerabilities across various content management systems (CMS) and their plugins. Threat actors are conducting mass scanning operations to identify vulnerable, internet-facing CMS installations and deploy webshells for persistent remote access. The campaign is notable for its scale — Five Eyes cyber agencies have flagged that AI-enabled tooling is compressing the window between vulnerability disclosure and active exploitation, meaning patch cycles now have far less margin than before.

What are the systems affected?

The campaign exploits **17 distinct CVEs** across CMS platforms and plugins, predominantly WordPress-based:

Simple File List (WordPress) - **CVE-2020-36847**

WavePlayer (WordPress) - **CVE-2025-12057**

BerqWP (WordPress - **CVE-2025-7443**

WPBookit (WordPress) - **CVE-2025-7852**

Ninja Forms (WordPress) - **CVE-2026-0740**

ThemeREX Addons (WordPress) - **CVE-2026-1969**

Breeze Cache (WordPress) - **CVE-2026-3844**

pay-uz (WordPress) - **CVE-2026-31843**

ACF Extended (WordPress) - **CVE-2025-13486**

Sneeit Framework - **CVE-2025-6389**

WPvivid Backup (WordPress) - **CVE-2026-1357**

Gravity Forms (WordPress) - **CVE-2025-12352**

GutenKit/Hunk Companion - Likely **CVE-2024-9234**

Craft CMS - **CVE-2025-32432**

MaxSite CMS - **CVE-2026-3395**

MetInfo CMS - **CVE-2026-29014**

Joomla JCE - **CVE-2026-48907**

What does this mean?

How it's exploited:

The vulnerability classes involved primarily fall into four categories:

1. **Unauthenticated file upload** – allows direct placement of malicious files (webshells) into web-accessible directories without any authentication.
2. **Remote code execution (RCE)** – allows arbitrary command execution on the underlying server.
3. **Server-side request forgery (SSRF)** – allows the attacker to make the server issue requests on their behalf, often to reach internal resources.
4. **Deserialisation flaws** – allow crafted serialized objects to trigger unintended code execution when processed by the application.

Attack sequence:

- Mass scanning identifies internet-facing CMS instances running vulnerable software/plugin versions.

- The specific vulnerability class (upload, RCE, SSRF, or deserialisation) is exploited to plant a webshell in the web directory or plugin directory.
- The webshell provides persistent command execution, which attackers use to:
    - Deface or disrupt the website
    - Harvest credentials or data submitted by site users
    - Stage further malware to compromise site visitors
    - Move laterally into the broader network from the compromised web server

Mitigation process

CERTVU recommends the following:

Detection / Immediate response:

1. Inspect the CMS web directory (and plugin directories, if a vulnerable plugin is confirmed) for abnormal or unexpected files.
2. Review web access logs for GET/POST requests to known webshell paths.
3. Treat any server with an identified webshell as compromised — isolate it and audit authentication/network logs for malicious activity.
4. Trace back through logs to identify the initial exploitation request.
5. Check edge firewall/network logs for communication with any known malicious IPs associated with the webshells.
6. Look for evidence of persistence, lateral movement, new account creation, exfiltration, or additional malware deployment.
7. Patch the vulnerable component, remove/quarantine webshells and persistence mechanisms, then verify clean before returning the host to production.
8. If compromise is confirmed, restore from a known-good backup.

Reference

1. https://www.cyber.gov.au/about-us/view-all-content/alerts-and-advisories/large-scale-exploitation-campaign-targeting-website-content-management-systems-cms
2. https://www.cisa.gov/news-events/bulletins/sb26-187
3. https://nvd.nist.gov/vuln/detail/CVE-2020-36847