



CIRT-ADV-2026-001 SECURITY ADVISORY

PHP2Shell: Critical PHP Vulnerabilities

CVE-2026-109618 | CVE-2026-109619

22 October 2026

Security Advisory: PHP2Shell Critical PHP Vulnerabilities

Eswatini Communications Commission | Issued 22 October 2026 | TLP:CLEAR

Advisory reference	CIRT-ADV-2026-001
Date issued	22 October 2026
Severity	Critical (CVSS v4.0: 9.2)
CVE IDs	CVE-2026-109618, CVE-2026-109619
Common name	PHP2Shell
Affected product	PHP (the PHP language runtime), versions 5.4 and later
Patch available	Yes: PHP 8.5.12, 8.4.27, 8.3.36, 8.2.35 (released 21 October 2026)
Exploitation	Active exploitation reported; public exploit code available
TLP	TLP:CLEAR (may be distributed without restriction)

Summary

The Eswatini Computer Incident Response Team (CIRT) urges all organisations running PHP-based websites and applications to update PHP immediately. Two critical vulnerabilities in PHP itself, together known as **PHP2Shell**, allow an attacker to run commands on a web server without logging in. They were disclosed on 21 October 2026 and attacks are already being seen worldwide.

Because the flaws are in the PHP engine, any PHP application can be affected, whatever the application. This includes government portals, banking and insurance customer portals, e-commerce sites, and content management systems such as WordPress, Joomla, Drupal and Magento.

Description and Impact

Both vulnerabilities are memory-corruption (use-after-free) flaws in the PHP core. When a request hits a fatal error, most often running out of memory, PHP can leave freed memory behind that a later request treats as live data. An attacker can shape that memory to run system commands.

The main attack path is PHP's **session upload progress** feature, which is switched on by default (`session.upload_progress.enabled = On`). A single crafted file-upload request is enough, and it is processed before the website's own code runs, so login pages and application controls do not stop it. On PHP 8.5 a second path uses fatal-error backtraces.

A successful attack can lead to:

- **Full server compromise:** the attacker runs commands with the web server's privileges, installs web shells or backdoors, and moves deeper into the network.
- **Data theft:** one of the flaws can also leak server memory, similar to Heartbleed. Leaked memory may contain passwords, session tokens, API keys and TLS private keys belonging to other users.
- **Service disruption:** defacement, ransomware deployment or outages of public-facing services.

Affected and Fixed Versions

All PHP versions from 5.4 onward are affected. Supported branches have a fix; end-of-life branches will not receive one.

PHP branch	Vulnerable versions	Fixed version
8.5	8.5.0 to 8.5.11	8.5.12
8.4	8.4.0 to 8.4.26	8.4.27
8.3	8.3.0 to 8.3.35	8.3.36
8.2	8.2.0 to 8.2.34	8.2.35
8.1 and 8.0	All	None (end of life): upgrade to a supported branch
7.x and 5.4 to 5.6	All	None (end of life): upgrade to a supported branch

Linux distributions (Red Hat, Debian, Ubuntu and others) may ship the fix as a patched package without changing the PHP version number. Check your vendor's security advisory as well as the version reported by `php -v`. Hosting control panels and appliances that bundle PHP (for example cPanel, Plesk, Pterodactyl) also need their own updates.

Exploitation Status

PHP2Shell is being actively exploited. Technical details leaked before the coordinated disclosure, and several working exploits, including ones that open a remote shell from a single URL, are publicly available.

- Cloudflare reported attack traffic against its customers on 21 October 2026.
- Security firm Hadrian reported exploitation at scale, mostly through the session upload progress path.
- Researchers confirmed unauthenticated remote code execution against Magento 2.4.9 and ten other PHP applications.

The CIRT considers any internet-facing PHP server that has not been patched since 21 October 2026 to be at high risk of compromise.

Recommended Actions

1. Patch immediately (the only complete fix)

1. Identify every server, container and hosting account running PHP, including test, staging and third-party hosted sites.
2. Update to PHP 8.5.12, 8.4.27, 8.3.36 or 8.2.35, or apply your Linux vendor's patched package. Update all PHP packages together, then restart PHP-FPM or Apache.
3. Confirm the running version with `php -v` and against the version actually serving the website (PHP-FPM or Apache module), not just the command-line PHP.
4. Plan urgent migration for servers on end-of-life PHP (8.1 and older). Until then, isolate them or place them behind a web application firewall.

2. Interim mitigations (only if you cannot patch today)

These settings reduce risk but **do not fully protect** the server. Apply them in `php.ini` or the PHP-FPM pool configuration and restart PHP:

```
session.upload_progress.enabled = 0
memory_limit = -1
; PHP 8.5 only:
fatal_error_backtraces = 0
```

For PHP-FPM pools, use `php_admin_flag[session.upload_progress.enabled] = off` and `php_admin_value[memory_limit] = -1`. Note that `memory_limit = -1`

lets one request use all server memory, so monitor resource use. Where a web application firewall is available, block requests containing the `PHP_SESSION_UPLOAD_PROGRESS` field.

3. Check for signs of compromise

- **Web server logs:** multipart POST requests containing `PHP_SESSION_UPLOAD_PROGRESS`, especially from unknown sources since 20 October 2026.
- **PHP error logs:** repeated out-of-memory or other fatal errors.
- **Session storage:** session files containing `upload_progress_` entries with unusual or code-like content.
- **Processes:** shells or unexpected programs started by the web server user (for example `www-data` or `apache`).
- **File system:** new or modified `.php` files in web directories (possible web shells).
- **Network:** unexpected outbound connections from web servers.

4. If you find signs of compromise

1. Isolate the affected server and preserve logs and disk images before cleaning.
2. Report the incident to the CIRT: info@nscirt.org.sz
3. Rotate all secrets the server held: database and application passwords, API keys, session secrets and TLS certificates and keys. Memory leakage means these may be exposed even without a confirmed shell.
4. Rebuild from a known-good image rather than cleaning in place, then patch before returning to service.

References

- [PHP 8.4.27 security release notes](#)
- [Sansec: PHP2Shell critical PHP RCE](#)
- [JFrog Security Research: PHP2Shell, all you need to know](#)
- [Upwind: PHP2Shell update with patch versions](#)
- [Brandefense: PHP2Shell fixed versions and exposure check](#)
- [PHP2Shell version checker \(open source\)](#)