CAA-2026-0025 Observed PHP2Shell Exploitation Activity

This report is distributed as **TLP:CLEAR**. Recipients may share this information without restriction. Information is subject to standard copyright rules.

Disclaimer | CyberAlberta

Summary

Between 21 October 2026 and 22 October 2026, CyberAlberta Threat Intelligence identified several unsuccessful PHP2Shell exploitation attempts.

Details

Vulnerability Details

PHP2Shell is the moniker for two actively exploited PHP vulnerabilities: CVE-2026-109618 and CVE-2026-109619. Both vulnerabilities enable unauthenticated remote code execution on vulnerable PHP versions, including versions after PHP 7.1 and PHP 5.4.[1] Emergency patches for PHP2Shell were released on 21 October 2026, available in PHP versions 8.5.12, 8.4.27, 8.3.36, and 8.2.35.[2]

Threat Actors Abusing Public Proof-of-Concept Exploit Code

In several instances, threat actors likely used publicly available proof-of-concept (PoC) code to exploit PHP2Shell vulnerabilities, as telemetry captured user-agents that matched the unique default string set in several PoCs.

The IP address `31[.]205.10.244` attempted exploitation using the `sudo3rs-PHP2Shell-Checker/1.4.0` user agent, consistent with a PHP2Shell vulnerability scanner made by GitHub user Masriyan.[3]

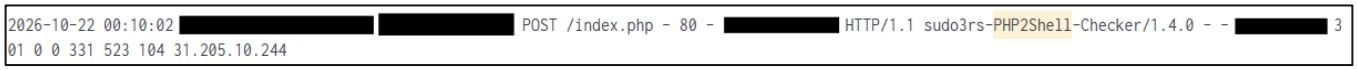

Figure 1. Example Malicious Web Request

The IP addresses `45[.]198.224.169` and `64[.]95.12.7` attempted exploitation using the `php2shell-109618` user agent, consistent with a PHP2Shell exploit made by the GitHub user Zer061, which has since been removed from GitHub.[4]

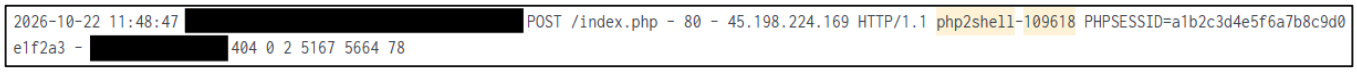

Figure 2. Example Malicious Web Request

The IP address `64[.]31.36.196` scanned multiple endpoints for vulnerabilities using the `Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+(KHTML,+like+Gecko)+Chr`

[1] https://www.crowdstrike.com/en-us/blog/crowdstrike-protects-customers-from-php-rce-zero-days/
[2] https://www.php.net/archive/2026.php#2026-10-21-5
[3] https://github.com/Masriyan/php2shell/blob/main/php2shell.py
[4] https://web.archive.org/web/20261021082048/http://github.com/Zer061/CVE-2026-109618/

`ome/60.0.3112.113+Safari/537.36+PHP2Shell/1.0.0` user agent, consistent with a PHP2Shell scanner made by GitHub user kgervot-an.[5]

Figure 3. Example Malicious Web Request

In several other cases, user agents signaling a similar intent (e.g., `php2shell-go/1.0`) were observed. However, analysts could not link them to known PoCs. It is likely that these represented similar, low sophistication exploitation, or scanning attempts.

Post Exploitation Activity

CyberAlberta Threat Intelligence has not observed post-exploitation activity; however, several open-source security blogs have detailed follow-on activity. Security researchers at Arctic Wolf identified post-exploitation activity including command execution, and second-stage payload delivery.[6] Security researchers at VulnCheck identified threat actors attempting to upload web shells.[7]

Assessment

CyberAlberta Threat Intelligence assesses with low confidence that the volume of PHP2Shell exploitation attempts will likely increase in the near term, as accurate details about the vulnerabilities continue to emerge from community research and PoC code reliability improves. As of 22 October 2026, multiple security vendors report exploitation attempts with no confirmed compromise,[8][9][10] likely attributable in part to the usage of non-functioning public PoC code.[11]

Recommendations

It is recommended that organizations perform the following actions:

- Identify all assets, including third-party systems, that have used vulnerable PHP components as of 17 October 2026.[12]
- Update PHP on all vulnerable assets to at least version 8.5.12, 8.4.27, 8.3.36, or 8.2.35, prioritizing public-facing assets. Apply patches for third-party systems as they are made available. Disable internet access for systems if patching is not possible.
- Conduct threat hunting activities across all identified assets.[13]

Indicators of Compromise (IOCs)

The following IOCs characterize PHP2Shell activity described in this report.

Description	Indicator
Source IP, Attempted PHP2Shell Exploitation	`31[.]205.10.244`
	`157[.]85.212.242`

[5] https://github.com/assetnote/php2shell-scanner/blob/master/scanner.py
[6] https://arcticwolf.com/resources/blog/php2shell-critical-php-vulnerabilities-enable-unauthenticated-remote-code-execution/
[7] https://www.vulncheck.com/blog/php2shell-vulnerabilities
[8] https://www.sophos.com/en-us/blog/proof-of-concept-exploit-released-for-php-vulnerabilities-109618-109619
[9] https://arcticwolf.com/resources/blog/php2shell-critical-php-vulnerabilities-enable-unauthenticated-remote-code-execution/
[10] https://aws.amazon.com/blogs/security/amazon-threat-intelligence-identifies-php2shell-exploitation-prior-to-public-disclosure-in-madpot/
[11] https://www.vulncheck.com/blog/php2shell-vulnerabilities
[12] https://aws.amazon.com/blogs/security/amazon-threat-intelligence-identifies-php2shell-exploitation-prior-to-public-disclosure-in-madpot/
[13] https://www.cyber.gc.ca/en/alerts-advisories/al26-026-critical-vulnerabilities-impacting-php-cve-2026-109618-cve-2026-109619

	79[.]127.134.9
	64[.]95.12.7
	188[.]166.108.219
	45[.]198.224.169
	85[.]11.167.243
	5[.]175.186.4
	34[.]79.202.206
	66[.]42.39.11
	64[.]31.36.196
	187[.]15.162.51
	182[.]10.137.120
User Agent	sudo3rs-PHP2Shell-Checker/1.4.0
	PHP2Shell-BatchScanner/1.0
	Mozilla/5.0+(php2shell-research)
	php2shell-109618
	wp-scan/0.2
	php2shell-scan/109618
	php2shell-research/1.0+(+security.txt+discovery;+one+request+per+domain)
	Mozilla/5.0+PHP2Shell-Scanner/1.0
	Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+(KHTML,+like+Gecko)+Chrome/60.0.3112.113+Safari/537.36+PHP2Shell/1.0.0
	shellflow-php2shell/1.0
	php2shell-go/1.0
Cookie	PHPSESSID=a1b2c3d4e5f6a7b8c9d0e1f2a3b4c5d6
	PHPSESSID=a1b2c3d4e5f6a7b8c9d0e1f2a3
	PHPSESSID=p2s_fixation_probe_12345
	PHPSESSID=lc07kvusc89qqrc3mkpv8svw4d

Table 1. PHP2Shell Indicators of Compromise