

EOTISEC

NATIONAL SECURITY INTELLIGENCE REPORT FOR BUSINESS

EOTISEC Analytical Division
DEFENDER APPENDIX

A1. DOCUMENT IDENTIFICATION

Document Title	PHP2Shell Defender Appendix: Detection, Hunt, and Mitigation Package
Companion To	EOTISEC-2026-063, PHP2Shell: Probability and Severity of Successful Exploitation of CVE-2026-109618 and CVE-2026-109619
Date	October 23, 2026
Classification	BUSINESS SENSITIVE // EOTISEC SUBSCRIBER DISTRIBUTION
Originator	EOTISEC Analytical Division
Prepared By	Sam Liles, PhD, EOTISEC Analytical Division
Reviewed By	Dr. Samuel Liles, CISO / Principal
Distribution	EOTISEC Subscriber Base

A2. SCOPE AND HOW TO USE THIS APPENDIX

This appendix is the operational companion to the EOTISEC PHP2Shell assessment. The main report answers the probability, severity, and trajectory question. This document gives defenders the indicators, detection logic, mitigations, and hunt procedures to act on it. It covers CVE-2026-109618 and CVE-2026-109619. All indicators were gathered from open reporting between October 17 and October 23, 2026 and verified against the originating source.

READ THIS FIRST. Every network indicator below is an *attempt* indicator, not a compromise indicator. As of October 23, no source has named a victim with post-exploitation evidence. Finding this traffic in your logs means you were scanned or probed, which is true of most of the internet. It does not mean you were breached. The host and filesystem artifacts in Section A5 are the ones that would indicate a successful plant. Treat the IP and user-agent lists as short-lived. The sources state the user-agents are trivial to change and the IPs rotate quickly, so hunt on behavior and durable artifacts first, and use the address lists for retrospective log searches rather than long-term blocklists.

A3. AFFECTED AND FIXED VERSIONS

Branch	Affected	Fixed release	Action
8.5.x	8.5.0 through 8.5.11	8.5.12	Update and restart all workers.
8.4.x	8.4.0 through 8.4.26	8.4.27	Update and restart all workers.
8.3.x	8.3.0 through 8.3.35	8.3.36	Update and restart all workers.
8.2.x	8.2.0 through 8.2.34	8.2.35	Update and restart all workers.
7.4, 8.0, 8.1	All	No upstream release	Use distribution or LTS vendor package, or the PHP team minimal source backports. Verify, do not assume.
5.4 through 7.3	All	None	End of life. No fix. Upgrade to a supported branch or isolate from public networks.

Version note. The CLI binary version can differ from the version that serves web requests. Check the serving version through phpinfo output or the running process, not php --version alone. Distribution packages often backport fixes

without changing the version string, so a version-string scan can both overcount exposure and miss patched hosts. Treat version scans as a starting inventory, not proof of state.

A4. EXPOSURE TRIAGE

Apache mod_php with default settings is the highest-risk configuration, because CVE-2026-109618 is reachable by an unauthenticated request before application code runs. PHP-FPM is reachable only through application code paths such as session start or unserialize. CrowdStrike gives a rough reachability condition for the mod_php path. The host is exposed when memory_limit is less than four times post_max_size. Their examples are post_max_size 64M with memory_limit 128M, and post_max_size 100M with memory_limit 256M. Remember that post_max_size can be set per directory in .htaccess or a virtual host, so individual sites on shared hosting can be exposed even when the global config is not.

Check the serving configuration on each host:

```
php -i | grep -E "(memory_limit|post_max_size|upload_max_filesize|Loaded Configuration File)"
# compare against the condition: memory_limit >= 4 x post_max_size avoids the mod_php
reachability window
# review per-directory overrides in .htaccess and vhost config, not only the global
php.ini
```

A5. INDICATORS

A5.1 Durable host and filesystem artifacts (highest value)

These come from a webshell plant attempt recorded by VulnCheck canaries and from tooling recovered by SOCRadar. Several values appear hard-coded in the tooling, which makes them good hunt terms. A hit here, unlike a network hit, points at an attempted or successful plant and warrants investigation.

Artifact	Value	Where to look
Execution marker	PWNED::	Any HTTP response body. Presence means the planted payload ran.
Planted payload	<?php echo 'PWNED::'; system(\$_GET['c']); ?>	Web root files, session files, request bodies.
Session file	sess_xqsuvwz3n97wu8j3emz1axuo5y	Session save path (default /var/lib/php/sessions or tmp).
PHPSESSID value	xqsuvwz3n97wu8j3emz1axuo5y	Request and response cookies, access logs.
Multipart boundary	-----p2s9rdme2st6guc9g0rafiC	Raw request captures, WAF logs.
Upload filename	p2s.txt	Upload temp dirs, web root, access logs.
Upload filename pattern	MARKER_<8 hex>.txt	Upload temp dirs (SOCRad tooling).
Screening session prefixes	fprint and dp	Session IDs during fingerprinting (SOCRad tooling).
Field name	PHP_SESSION_UPLOAD_PROGRESS	Multipart POST bodies. Legitimate but central to the 109619 path, so correlate with anomalies.
Read-attempt path	php://filter to download.php	Access logs, following an upload POST.

A5.2 File hashes (SHA-256)

Uploaded multipart parts and follow-up body captured by VulnCheck canaries.

Part	Size	SHA-256
Part a	10,370 B	150ae88f0999032e2a6113501d6082f54a3ed98d02427b21a96e773a27d7bf16
Part a	8,922 B	40caf199calebe056a74427f8e9db06c66899e33ae09c0ec2769ec70bf6dd8e7
Part b	64 B	27d926a6708d6769d97d065e7a5a860ab496eab272c03d1cebc55ec71b954fa2
Body	13 B	1485e14e0f62bc94664eec341cabd275dad7d30b1d496c40ec340c9a0d9f0ba8

A5.3 Scanner and exploit user-agents

Trivial to change. Use for retrospective log search, not durable blocking. Match on the substring php2shell rather than exact strings.

```
php2shell-scan/109618
php2shell-109618
php2shell-scan/1.3 (eigen-oppervlak controle; CVE-2026-109618/109619)
php2shell-verify/1.0
Mozilla/5.0 (php2shell-poc)
```

Scanner TLS fingerprint. Hunt.io published a JA4H fingerprint for one scanner:

A5.4 Network indicators

Grouped by role. Investigate and retrospectively search rather than permanently block, with the exception of the botnet payload and C2 infrastructure, which is reasonable to block outright.

Role	Indicator	Source and note
Scanner / probe IP	85.11.167.243	AS209630, NL. Hit 188 canary hosts (VulnCheck).
Scanner / probe IP	45.198.224.169 45.198.224.0/24	AS215925, VPSVAULT. Scanner network and host (VulnCheck).
Scanner / probe IP	109.71.252.97	AS58087/AS213250, DE (VulnCheck).
Exploit-attempt IP	221.14.159.134	China Unicom, Henan. Webshell plant attempt, failed (VulnCheck).
Research checker IP	91.90.123.25	M247, BE. UA php2shell-verify/1.0 (VulnCheck, Hunt.io).
Pre-disclosure attempt	103.216.220.92	First MadPot hit, Oct 17 (AWS).
Attempt IP	31.223.184.166	Oct 21 (AWS MadPot).
Attempt IP	37.19.200.5	Oct 21 (AWS MadPot).
Attempt IP	134.209.159.64	Oct 21 (AWS MadPot).
Attempt IP	103.114.163.188	Oct 21 (AWS MadPot).
Attempt IP	95.217.98.37	Oct 21 (AWS MadPot).
Botnet payload host	92.118.112.100	Potassium loader source. Block (THN, Hunt.io).
Botnet binary host	89.32.41.16 89.32.41.50 45.198.224.88	Potassium binaries. Block (Hunt.io).
Botnet C2 domains	*.botlesscucks.st *.godofthunder.st *.gettingheadfromduckysmom.st	Block. thunder.godofthunder.st to 134.199.154.64, C2 TCP 50792 (Hunt.io, THN).
DDoS targets	45.88.138.0 45.88.138.44	Potassium flood targets, one Ukrainian network (THN).

Attacker staging and tooling hosts. Recovered open directories and operator servers. No confirmed victims, but worth retrospective outbound and inbound log searches:

```
206.188.197.105 (SOCRadar operator server; open ports 9999, 8899, 8888)
102.210.149.166 122.10.110.72 122.10.17.49 58.221.60.160
```

41.223.30.7	157.230.117.70	104.219.251.31	107.150.31.151
217.160.132.82	47.100.125.108	54.169.109.7	139.162.165.136

A6. DETECTION LOGIC

A6.1 Web server and PHP log signature (primary)

Exploitation of CVE-2026-109618 drives the worker to a memory-exhaustion fatal during request startup. The distinguishing log line pairs a memory-exhausted error with a location of Unknown on line 0, which means no PHP file was executing when memory blew. Responses typically return HTTP 200 and the application log shows nothing. Alert on bursts of this pattern from October 17 forward.

```
# error log pattern (grep / SIEM regex)
Allowed memory size of [0-9]+ bytes exhausted.*in Unknown on line 0

# supporting indicators to correlate
- bursts of HTTP 500 followed by 200 on the same endpoint
- unusually large or malformed PHPSESSID response cookies
- unexplained PHP worker crashes or respawn churn
- multipart POSTs with very many parts or empty filenames, or no Content-Length
```

A6.2 Process lineage (confirms a shell)

A successful code-execution outcome spawns a child process from the web server or PHP worker. This is the strongest evidence of compromise and the detection that matters most, since application logs stay quiet.

```
Parent:  php-fpm | php-cgi | php-cgi.exe | httpd | apache2 | litespeed | frankenphp
Child:   sh | bash | dash | cmd.exe | powershell.exe | python | perl
        curl | wget | nc | ncat | certutil.exe
Cmdline: /dev/tcp/ | fsockopen | mkfifo | base64 -d | chmod +x
Recon:   whoami | id | uname -a | hostname | ipconfig | ip a
```

A6.3 Sigma rule, process execution

Portable starting point. Tune parent and child lists to your environment and reduce false positives from legitimate PHP that shells out (some CMS plugins, cron wrappers, image tooling).

```
title: PHP or Web Server Spawning a Shell or Download Tool (PHP2Shell)
id: eoti-php2shell-proc-001
status: experimental
description: Possible post-exploitation of CVE-2026-109618 / CVE-2026-109619
logsource:
  category: process_creation
detection:
  parent:
    ParentImage|endswith:
      - '/php-fpm'
      - '/php-cgi'
      - '/httpd'
      - '/apache2'
      - '\\php-cgi.exe'
  child:
    Image|endswith:
      - '/sh'
      - '/bash'
      - '/python'
      - '/perl'
      - '/curl'
      - '/wget'
      - '/nc'
      - '\\cmd.exe'
      - '\\powershell.exe'
  condition: parent and child
level: high
```

```
falsepositives:
- Legitimate PHP applications that invoke shell utilities
```

A6.4 Web log and WAF query, request side

```
# access-log / WAF hunt (pseudo)
( http.user_agent contains 'php2shell' )
OR ( http.request.method == 'POST'
    AND http.request.body contains 'PHP_SESSION_UPLOAD_PROGRESS'
    AND http.request.body.length > <your_normal_upload_ceiling> )
OR ( uri contains 'php://filter' )
OR ( http.request.body contains '-----p2s9rdme2st6guc9g0rafiC' )
```

A6.5 WAF and edge filtering

Stopgap only. Attackers are mutating payloads using feedback from blocked requests, so treat the edge as time bought for patching, not a fix. Apply request-shape limits rather than signature matching where possible:

- Cap multipart part count, per-part header count, field-name nesting depth, and total decoded body size at the proxy.
- Enforce these limits on requests with no Content-Length and on chunked or HTTP/2 multipart, which are the bypass vectors.
- If the application accepts no uploads, block multipart/form-data outright at the edge.
- Managed WAF rules from major CDNs went live around disclosure. Enable them, and do not assume they catch mutated payloads.

A7. INTERIM MITIGATIONS (NOT A FIX)

Patching and restarting all workers is the only complete remediation. The settings below reduce exposure on systems that cannot patch immediately. They address specific vectors and the PHP team states they are not a complete solution. Apply them at php.ini, FPM pool, or server config level, because ini_set() and .user.ini are not sufficient, then restart every worker.

A7.1 CVE-2026-109618 (OOM corruption)

- `memory_limit = -1` set where the application cannot override it. Counterintuitive but correct, since a hard limit is the trigger. Note a failed underlying allocation can still reach the path, so this is partial.
- `post_max_size = 8M` if no uploads, or `32M` if uploads are needed, with `upload_max_filesize` kept below it.
- Do **not** use `RLIMIT_AS` or `RLIMIT_DATA`. They cause the allocation failures that trigger the bug.
- A container or cgroup `memory.max` can bound availability impact, but is not a mitigation for the corruption.

A7.2 CVE-2026-109619 (stale global state)

- `session.upload_progress.enabled = 0` then restart all workers. Keep it off even after patching if unused.
- On PHP 8.5, `fatal_error_backtraces = 0` then restart all workers.
- Remove per-request settings that emit diagnostics before init, including deprecated or invalid INI under `.htaccess` `php_value` and FPM `PHP_VALUE`. Suppressing with `@` or `error_reporting` does not help.
- Reduce attacker-reachable fatal errors and treat execution timeouts as fatal. Consider `pm.max_requests = 1` on FPM pools to retire workers between requests.

A7.3 Hardening around the observed tooling

- `session.use_strict_mode = 1` , `expose_php = Off` , and remove public `phpinfo` pages, which the fingerprinting scripts look for.

A8. HUNT AND TRIAGE WORKFLOW

- 1. Set the lookback to October 17, 2026.** The earliest honeypot hit predates the patch by four days, so start every search there, not at the disclosure date.
- 2. Confirm the serving version and patch state per host.** Use the serving version, not the CLI. Record which hosts were exposed and for how long.
- 3. Search durable artifacts first (A5.1 and A5.2).** The `PWNED::` marker, the session filename, the boundary, the payload string, and the hashes. A hit here is actionable. A network-only hit is not.
- 4. Run the log signature (A6.1).** Look for the memory-exhausted plus Unknown on line 0 pattern in bursts, and correlate with worker churn and malformed session cookies.
- 5. Run process lineage (A6.2, A6.3).** Any web server or PHP worker spawning a shell, interpreter, or download tool in the window is a high-priority lead.
- 6. Scope outbound.** Check for fetches of ELF or Go loaders from PHP processes and for connections to the botnet infrastructure in A5.4.
- 7. Decide on exposure, not just evidence.** If a host served an unpatched, reachable configuration during the window, assume memory disclosure may have leaked secrets even with no shell evidence, and move to the response steps.

A9. IF COMPROMISE OR EXPOSURE IS CONFIRMED

- Preserve first. Capture web server and PHP logs, config snapshots, session directories, and suspicious files before changing anything.
- Isolate the host from public networks. Rebuilding from a known-good image is preferred over cleaning in place.
- Rotate every secret the web process could reach. Database and API credentials, session secrets, signing keys, and any certificate private keys readable by the worker. Then invalidate active sessions.
- Review what those credentials can reach, since memory disclosure can expose secrets belonging to other requests and other tenants on the same worker.
- Hunt for persistence. New web root files, cron jobs, scheduled tasks, and accounts tied to the web server or PHP processes.
- Patch before returning to service, and restart all workers. Do not rely on the interim settings as the end state.

A10. SOURCE NOTE

Artifacts and indicators in this appendix are drawn from VulnCheck canary and exploit reporting, CrowdStrike detection guidance, Hunt.io open-directory analysis, SOCRadar attacker-tooling analysis, Amazon MadPot telemetry, The Hacker News reporting on the Potassium botnet, and the PHP Foundation and GitHub Security Advisory mitigation guidance. Full citations are in Section 10 of the companion report EOTISEC-2026-____. Indicators are current as of October 23, 2026. Network indicators are expected to rotate. No classified sources were used.