

PHP2Shell

CVE-2026-109619 and CVE-2026-109618:
detection and threat hunting in GREYCORTEX Mendel

9.2 CVSS v4.0, both CVEs	RCE Unauthenticated, before application code	7 IDS rules for Mendel, one helper	8.5.12 ■ 8.4.27 8.3.36 ■ 8.2.35 Fixed PHP versions
------------------------------------	--	--	--

1 What the vulnerability is

	CVE-2026-109618	CVE-2026-109619
Class	PHP runs out of memory while setting up the request. Cleanup then uses half-initialised request data (OOM via <code>longjmp()</code>).	Request-owned state outlives its freed memory. A later request on the same worker uses it (cross-request use-after-free).
Known triggers	Multipart body parsing; on <code>mod_php</code> also a normal urlencoded POST	Session upload-progress handler (<code>PHP_SESSION_UPLOAD_PROGRESS</code> form field); fatal-error backtraces on PHP 8.5; warnings from per-request <code>php_value</code> / <code>PHP_VALUE</code>
Impact	Unauthenticated RCE (<code>mod_php</code> confirmed)	Unauthenticated heap UAF, memory disclosure (other users' passwords, tokens, keys), RCE
Severity	CVSS v4.0 9.2	CVSS v4.0 9.2

- **Runs before application code.** This is not a vulnerability in logins, CSRF checks or WAF. Security measures inside the app do not stop it.
- **Affected:** every SAPI: `mod_php`, PHP-FPM, `php-cgi`, FrankenPHP, LiteSpeed. `mod_php` is the highest risk.
- **Fixed in:** 8.5.12, 8.4.27, 8.3.36, 8.2.35.
 - PHP published source backports for 7.4 / 8.0 / 8.1. Otherwise old branches depend on the distribution vendor.
- **Status on 2026-10-21:**
 - watchTowr honeypots and Cloudflare saw exploitation that day (Cloudflare from 07:43 UTC).
 - Magento 2.4.9 and 10 other apps have confirmed RCE.
 - No public IOCs yet.

Sources disagree on which CVE covers upload-progress.

Sansec maps it to -109618; Mattias Geniar, Huntress and Unit 42 map it to -109619. The rules below cover both CVEs, so the mapping does not change detection.

Network-visible signals

1 A multipart POST carrying `PHP_SESSION_UPLOAD_PROGRESS` .

2 Many multipart requests from one client that end in HTTP 5xx: `memory_limit` fatals, or workers crashing behind nginx (502/503).

3 A malformed or oversized session cookie in a response. This is the memory leak itself.

4 After RCE: the web server opens outbound connections, downloads with `curl/wget`, runs a reverse shell or sends `id` output.

2 IDS rules

sid	Event name (msg)	What it means	Mendel sev.
900047041	Exploit: PHP2Shell CVE-2026-109619 session upload-progress field in multipart POST	A client sent the 109619 trigger field. Legitimate use (upload progress bars) is rare.	8
900047042	Helper: PHP2Shell multipart POST seen (noalert)	Marks flows that carry multipart POSTs, for 043. Creates no events.	-
900047043	Exploit: PHP2Shell CVE-2026-109618/109619 repeated 5xx after multipart POST	One client got ≥10 responses of 5xx in 60 s on multipart flows. This is the grooming or leak loop.	8
900047044	Exploit: PHP2Shell CVE-2026-109619 5xx after upload-progress multipart POST	The trigger field was followed by a server error, so the trigger reached a fatal error.	8
900047045	Exploit: Possible memory disclosure - malformed or oversized session cookie in Set-Cookie	The server answered with a session id outside [A-Za-z0-9,-] or longer than 256 characters. This means leaked memory.	8
900047046	Policy: Web server outbound HTTP with download-tool User-Agent (possible PHP2Shell post-exploitation)	<code>\$HTTP_SERVERS</code> fetches from the internet with curl, Wget, python-requests, Go, Perl or PHP.	8
900047047	Exploit: Web server sent id command output (possible PHP2Shell RCE)	Output such as <code>uid=33(www-data)</code> <code>gid=33(www-data)</code> from a web server, in a webshell response or a reverse shell.	8

3 Threat-hunting playbook

Each step gives a filter block. Enter each line into the matching field of the **left sidebar filter** and click **Filter**. Fields that are not listed stay empty.

Alternatives

| separates alternatives inside one field, as Mendel expects.

Date range first

Since the PHP version was exposed, at least from 2026-10-14 (the leak).

Reuse

To reuse a filter, save it with Filter manager.

STEP 0 Which PHP servers you have (exposure)

1. List the web servers. These are the `$HTTP_SERVERS` that rules 046 and 047 use.

MENDEL FILTER	
View	Network > Hosts
Host tag	Role/Server/WEB

- Find which of them the internet can reach. Patch those first.

MENDEL FILTER	
View	Network > Analysis > Flows statistics
Host tag	Role/Server/WEB
Traffic	INBOUND
Attributes	Dst IP, Dst port
Metrics	Flows (Sum)

- Mendel cannot tell you the PHP version. The HTTP parser does not keep X-Powered-By , and TLS hides it. Get versions from the hosts (`php -v` , package lists).

STEP 1 Exploitation attempts (IDS events)

- Show all PHP2Shell IDS events.

MENDEL FILTER	
View	Events
Detection methods	IDS
Traffic	ALL
Event	PHP2Shell

- Read the events by sid:

- 041 /** The trigger was sent (and reached a fatal error). The source is the attacker and the
- 044** destination is the victim server.
- 043** An exploitation loop. Compare the count to 10 per minute, and check whether the same source hits several servers.
- 045** **Memory left the server.** Treat it as a confirmed leak: rotate passwords, sessions, API keys and TLS keys used by that PHP worker.
- 046 /** Post-exploitation. Go to step 3 now.
- 047**

- Pivot on the attacker: find scans and brute force before the attack, and other targets.

MENDEL FILTER	
View	Events
Detection methods	all
Host	<attacker IP>
Traffic	ALL

4. Pivot on the victim. Sort by Src packet size: large inbound uploads from one source just before 5xx errors match the attack shape.

MENDEL FILTER	
View	Network > Flows
Host	<victim server IP>
Service	dst:80 dst:443
Traffic	INBOUND

STEP 2 Attempts you cannot see in the payload (HTTPS, no decryption)

1. Look for one source with many flows and large uploads (high Src packet size) but tiny answers (low Dst packet size). That shape is an error page or a crashed worker. Compare with the same window on earlier days.

MENDEL FILTER	
View	Network > Analysis > Flows statistics
Host	<web server IP> (or Host tag: Role/Server/WEB)
Service	dst:443
Traffic	INBOUND
Attributes	Src IP, Dst IP
Metrics	Flows (Sum), Src packet size (Sum), Dst packet size (Sum)

2. Check Mendel's NBA events for the server.

MENDEL FILTER	
View	Events
Host	<web server IP>
Event	Outlier High Number of Connections Flows at Host

STEP 3 Post-exploitation on the web server

1. Find web-server post-exploitation events from existing ET/GCX content: webshells, phpshell, tool User-Agents, Tor, miners, dynamic DNS and outbound data anomalies.

MENDEL FILTER	
View	Events
Detection methods	all
Host tag	Role/Server/WEB
Traffic	ALL
Event	WebShell phpshell Attack-response Reverse Web Shell Python-urllib python-requests curl Wget Tor miner Coin DYNAMIC_DNS blacklist Outgoing data Scan Brute High Number of Connections

2. Show where the web servers connect outbound. Use Analysis: Network > Flows does not accept a host tag alone.

MENDEL FILTER	
View	Network > Analysis > Flows statistics
Host tag	Role/Server/WEB
Traffic	OUTBOUND
Attributes	Src IP, Dst IP, Dst country, Dst port
Metrics	Flows (Sum), Src packet size (Sum), Dst packet size (Sum)

3. In the result, flag:
- destinations or countries that were not there in the days before the attack;
 - ports such as 4444, 1337, 8080-9999, 53/UDP to non-resolvers, 6667;
 - long-lived flows with small packets in both directions (an interactive shell);
 - large Src packet size to one destination (exfiltration of a DB dump or source code).
4. Drill into each suspect destination. Note the start time, duration and bytes.

MENDEL FILTER	
View	Network > Flows
Host	<web server IP>
Country	<destination country> (or ASN: <destination ASN>)
Traffic	OUTBOUND

STEP 4 Lateral movement from the web server

1. List the internal destinations. Look for SSH, SMB, RDP, DB ports (3306, 5432, 1433, 6379) or Kubernetes / cloud metadata endpoints that this server never used before.

MENDEL FILTER	
View	Network > Analysis > Flows statistics
Host	<web server IP>
Traffic	INTERNAL
Attributes	Dst IP, Dst port
Metrics	Flows (Sum)

2. Check for internal reconnaissance and credential events.

MENDEL FILTER	
View	Events
Host	<web server IP>
Traffic	ALL
Event	Scan Sweep Brute SMB NTLM Kerberos SSH

STEP 5 Abuse of leaked credentials (after 045, or after any confirmed RCE)

The leak gives attackers session cookies, passwords and API keys. Watch for their later use.

1. Admin logins from new sources: look for source IPs and countries new to admin or back-office use.

MENDEL FILTER	
View	Network > Flows
Host	<web server IP>
Service	dst:443
Traffic	INBOUND

2. The app's backend contacted directly. A source other than the app server means leaked DB or API credentials.

MENDEL FILTER	
View	Network > Analysis > Flows statistics
Host	<DB / API / cache host IP>
Traffic	INTERNAL
Attributes	Src IP, Dst port
Metrics	Flows (Sum)

3. Third-party APIs called from unknown hosts.

MENDEL FILTER	
View	Network > Analysis > Flows statistics
ASN	<SaaS / payment / cloud provider ASN>
Attributes	Src IP, Dst IP
Metrics	Flows (Sum)

STEP 6 When something is confirmed

1. Isolate the server.

2. Keep the evidence:
 - the Mendel pcap of the flows (Packet capture);
 - web-server and PHP-FPM logs (5xx, Allowed memory size ... exhausted , worker exited on signal);
 - the process tree.
3. Patch PHP to a fixed version. Until then, apply the interim mitigations from the references:
 - block multipart at the web server;
 - set `session.upload_progress.enabled = 0` if the application does not depend on upload-progress tracking.
 - on PHP 8.5, `fatal_error_backtraces = off` ;
 - move per-request `php_value / PHP_VALUE` into `php.ini`;
 - `pm.max_requests = 1` .
 - lower `post_max_size` to a value compatible with your application's upload requirements. This also constrains file uploads, so test legitimate upload flows.
 - lower `max_input_nesting_level` where the application permits it. Complex forms may require a higher limit.
 - set `display_errors = 0` in production and restart the affected workers to prevent PHP error output from being exposed to remote users.
4. Rotate everything the PHP workers held in memory.

4 Caveats and limits

- **HTTPS is the main gap.** Rules 041-046 read HTTP content. Most PHP sites are HTTPS-only, so the IDS sees nothing unless the sensor monitors decrypted traffic: behind a TLS-terminating load balancer, reverse proxy or CDN-to-origin plain-HTTP leg. Over TLS only steps 2-5 work (flow shape, outbound and lateral connections, NBA), plus rule 047 for a plain-TCP reverse shell.
- **HTTP/2 / HTTP/3:**
 - The rules also cover HTTP/2, but HTTP/2 is almost always TLS.
 - QUIC/HTTP/3 is not inspectable.
- **Request bodies:**
 - Rule 041 finds the field only within the part of the request body the IDS inspects (the configured body limit). An attacker can put megabytes of parts before the field.
 - Rule 041 also misses a site that renamed `session.upload_progress.name` .
- **No PoC means no exact signature.** The rules match the trigger surface and the side effects, not the exploit bytes. Expect false positives from:

- rule 041: real upload progress bars, so check the page;
 - rule 043: buggy upload endpoints or overloaded servers;
 - rule 046: cron jobs and updates with curl or wget, so whitelist by destination.
- Tune with Mendel's false-positive rules, not by editing the rules.
- **Silent exploitation is possible:**
 - A successful leak that does not use the session cookie, or RCE in one request, creates no 5xx and no 045 event.
 - PHP-FPM restarts workers, so 502s may appear only at the nginx front.
 - The memory leak itself leaves no file or process (Mattias Geniar); rotating secrets is the only safe answer.
 - **Session cookie name:** rule 045 looks for the "sess" string in the cookie name. Apps with their own session name, such as Laravel and some other setups, should be covered. However, some session cookies with specific names require copies of this signature with the correct session cookie names.
 - **\$HTTP_SERVERS :** rules 046 and 047 depend on UnTE tagging the server `Role/Server/WEB` . A new or rarely used server may lack the tag, and then those rules do not see it. Rules 041-045 use `any` , so they cover every host.
 - **Sensor placement:** Mendel sees only mirrored traffic. Servers reached through another path, cloud workloads and internal east-west traffic on unmirrored segments are blind spots.
 - **Retrospective hunting:**
 - IDS events exist only from the deploy time on.
 - For earlier dates use the flow-based steps 2-5, back to Mendel's flow retention.
 - The full-JSON HTTP L7 data (Method, Status) is sampled, so absence there proves nothing.
 - **Mendel UI:**
 - Network > Flows needs a Host, Subnet, MAC, Service, Protocol, ASN or Country in the filter. A Host tag alone is rejected.
 - Network > Analysis accepts a Host tag alone.

5

References

1. PHP Group, security advisory "Critical remote code execution vulnerabilities, update now", 2026-10-21
<https://www.php.net/archive/2026.php#2026-10-21-5>
2. PHP GitHub advisories GHSA-w8hw-w9fx-j79w and GHSA-6wmf-4rv5-p3rp
<https://github.com/php/php-src/security/advisories>
3. SOC Prime, "CVE-2026-109618 & CVE-2026-109619 analysis"
<https://socprime.com/blog/cve-2026-109618-cve-2026-109619-analysis/>
4. Sansec, "PHP2Shell: Critical PHP RCE"
<https://sansec.io/research/php2shell>
5. Mattias Geniar, "PHP CVE-2026-109618 and CVE-2026-109619: early mitigation steps"
<https://ma.ttias.be/php-cve-2026-109618-early-mitigation/>

6. Huntress, "Rapid Response: php2shell"
<https://www.huntress.com/blog/php2shell-leaked-critical-php-vulnerabilities>
7. Palo Alto Unit 42, "Threat Brief: Critical Vulnerabilities in PHP"
<https://unit42.paloaltonetworks.com/cve-2026-109618-cve-2026-109619/>
8. watchTowr, "PHP RCE FAQ: PHP2Shell"
<https://watchtowr.com/intelligence/php-cve-2026-109618-faq/>
9. Brandefense, "PHP2Shell: PHP RCE"
<https://brandefense.io/blog/php2shell-php-rce-cve-2026-109618-109619/>
10. SecureLayer7, "PHP RCE CVE-2026-109618 & 109619: What To Do Now"
<https://blog.securelayer7.net/php-rce-cve-2026-109618-109619/>
11. Plesk KB, "Remote code execution in PHP shipped with Plesk"
<https://support.plesk.com/hc/en-us/articles/44397729587351>
12. HeroDevs, EOL fix for CVE-2026-109619
<https://www.herodevs.com/vulnerability-directory/cve-2026-109619>
13. Wiz, "PHP2Shell — AI uncovers critical vulnerabilities in the backbone of the web"
<https://www.php2shell.com/>