

EOTISEC

NATIONAL SECURITY INTELLIGENCE REPORT FOR BUSINESS

EOTISEC Analytical Division

SECTION 1: REPORT IDENTIFICATION

Report Title	PHP2Shell: Probability and Severity of Successful Exploitation of CVE-2026-109618 and CVE-2026-109619
Report Number	EOTISEC-2026-063
Date of Report	October 23, 2026
Classification	BUSINESS SENSITIVE // EOTISEC SUBSCRIBER DISTRIBUTION
Originator	EOTISEC Analytical Division
Subject	Assessment of the probability, severity, and trajectory of successful exploitation of the two PHP core remote code execution vulnerabilities known collectively as PHP2Shell, and the actions subscriber organizations should take.
Customer Sector	Financial Services; Critical Infrastructure and Government Services; Healthcare; Maritime Commerce; Technology and Web Hosting Providers
Prepared By	Sam Liles, PhD, EOTISEC Analytical Division
Reviewed By	Dr. Samuel Liles, CISO / Principal
Distribution	EOTISEC Subscriber Base

SECTION 2: SCOPE AND PURPOSE

This report assesses the probability, severity, and likely trajectory of successful exploitation of CVE-2026-109618 and CVE-2026-109619, the two PHP core remote code execution flaws disclosed on October 21, 2026 and known together as PHP2Shell¹. It answers one question. How likely is widespread successful compromise of subscriber systems in the near term, how bad would it be, and what should subscribers do now. The source base is open reporting gathered between October 17 and October 23, 2026. That base includes the PHP Foundation advisory and the two GitHub Security Advisories, the research write-ups from Calif, Wiz, and CrowdStrike, telemetry and exploit reporting from Amazon, VulnCheck, Cloudflare, Hunt.io, SOCRadar, Recorded Future, and watchTowr, and official statements from the United States, the United Kingdom, the Netherlands, Germany, Japan, and Canada. The report does not resolve the disputed earliest-affected version range, and it does not assess the separate matter of the coordinated disclosure failure except where that failure changed the exploitation timeline. All source material is open source. No classified sources were used.

SECTION 3: KEY JUDGMENTS

Each judgment below is a primary analytical conclusion drawn from the source material. ICD 203 probability language applies throughout. Confidence level and explanation follow each judgment. A Business Impact statement follows each judgment.

KEY JUDGMENT 1: Mass, reliable, internet-wide successful exploitation that compromises patched or default-configured modern PHP systems is **unlikely** in the near term. As of October 23, no source has named a single victim with post-exploitation evidence, and every public proof of concept tested against real x86-64 targets has failed without a second weakness or absent address space layout randomization.

Confidence: Moderate. Multiple independent vendors report large volumes of attempts and zero confirmed successful compromises²³. The mechanism that makes exploitation hard, the need to defeat randomization and to tune memory pressure per build, is well understood. The gap is that absence of reported success is not proof of absence, and commercial and criminal capability is improving.

BUSINESS IMPACT: *Supported-branch sites on current patches that applied the October 21 update, and modern default installs behind address space randomization, face low near-term risk of shell compromise. Financial Services, Healthcare, and Technology subscribers that patched should treat the shutdown wave as a cost event, not a breach event, and should resist pressure to keep internet-facing services offline longer than patch verification requires.*

KEY JUDGMENT 2: Targeted successful compromise of a subset of exposed systems is **very likely** over the near term, concentrated on end-of-life, embedded, and shared-hosting PHP that patches slowly or never. Roughly 42 percent of observed PHP hosts run end-of-life branches, and PHP 5.4 through 7.3 receive no fix at all.

Confidence: Moderate. Exposure counts for end-of-life PHP are corroborated across VulnCheck, Modat, and VirusTotal⁴⁵. Criminal tooling already targets old builds. The judgment that some of these will be compromised rests on analytical inference about attacker behavior and the long tail, not on a confirmed breach, so confidence is held at Moderate.

BUSINESS IMPACT: *Maritime Commerce, Critical Infrastructure, and smaller Web Hosting subscribers carry the real exposure here. Legacy PHP sits inside vendor appliances, control panels, billing systems, and port and logistics software that patch on vendor timelines. Subscribers should inventory every PHP runtime that serves web requests, including bundled and embedded copies, and isolate anything on 7.3 or older that cannot be upgraded.*

KEY JUDGMENT 3: PHP2Shell will **almost certainly** settle into a long-tail pattern closer to Log4Shell than to a Y2K-style non-event, with compromises surfacing over months rather than a single disclosure-week spike.

Confidence: Moderate. The pattern is consistent across the exposure data, the end-of-life concentration, and independent forecasts⁶. The probability term is high because the structural conditions, a huge install base and a slow-patching tail, are observable facts. Confidence is Moderate because the timing and scale of the tail cannot be observed directly this early.

BUSINESS IMPACT: *All sectors should plan for a multi-month exposure window, not a one-week emergency. Subscribers should keep PHP2Shell on the active risk register through at least the first quarter of 2027 and schedule recurring hunt sweeps rather than treating the patch as closure.*

KEY JUDGMENT 4: Memory disclosure, not shell access, is **likely** to be the first source of real-world harm. The disclosure path leaks process memory, which can expose passwords, session tokens, API keys, and certificates from other requests, and it is more forgiving to exploit than full code execution.

Confidence: Moderate. Wiz describes an application-agnostic memory disclosure that works without a session, and VulnCheck confirms a working leak on both x86-64 and ARM64 even where reliable code execution does not yet follow⁷. The inference that disclosure precedes shell harm is analytically sound but not yet backed by a named credential-theft incident, so confidence is Moderate.

BUSINESS IMPACT: *Financial Services, Healthcare, and Technology subscribers should assume that any pre-patch exposure window may have leaked secrets and should rotate credentials, API keys, and session secrets*

reachable by internet-facing PHP, then invalidate active sessions. Breaches that later trace back to stolen secrets will often not be attributed to PHP2Shell at all.

KEY JUDGMENT 5: A weaponized x86-64 remote code execution exploit that defeats randomization under production settings has **very likely** already reached capable actors, and broader availability is likely within the near term. VulnCheck has such an exploit in commercial distribution now, and criminal address-randomization tooling has already been recovered from an attacker server.

Confidence: Moderate. VulnCheck states it chained a memory leak to code execution on PHP 8.4.10 through 8.4.26 with randomization on and error display off, and SOCRadar recovered a lab-built one-click chain from a financially motivated operator⁸⁹. The step from commercial and lab capability to broad criminal use is an inference about diffusion, not an observed event, so confidence is Moderate.

BUSINESS IMPACT: *The window in which only patched systems are safe is closing. Every sector should treat version-current patching, not configuration workarounds, as the only durable control, and should assume that the public exploit situation on October 23 understates real capability in private hands.*

SECTION 4: SITUATION AND BACKGROUND

The two vulnerabilities

PHP2Shell is a pair of memory-safety flaws in the PHP core. CVE-2026-109618 is remote code execution through out-of-memory corruption. When PHP hits a memory limit at the wrong moment, it exits the request using a longjmp that does not unwind state, and shutdown code then reads or branches on request data that was only partly built¹⁰. CVE-2026-109619 is remote code execution through stale global state. A fatal error can leave global error, session, object, or extension state pointing at memory that belonged to a finished request, and a later request on the same persistent worker can then use that dangling reference, a use-after-free¹¹. Both carry a CVSS v4.0 base score of 9.2, with the vector CVSS:4.0/AV:N/AC:H/AT:P/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N. The attack complexity is High and attack requirements are Present, which matters for the analysis below. An early leaked score of 10.0 was wrong, and a member of the PHP security team corrected it in public¹².

The practical exposure differs by server configuration. On Apache with mod_php and default settings, CVE-2026-109618 is reachable by an unauthenticated request before any application code runs, and the PHP Foundation and the advisory both report a full working exploit under default limits¹⁰. PHP-FPM is confirmed exploitable only through application-level code paths such as session start or unserialize, not before application code. CrowdStrike frames the mod_php exposure as a rough condition, that memory_limit is less than four times post_max_size, and gives example configurations that meet it⁹. For CVE-2026-109619 the advisory names two confirmed consumers of stale state. The PHP 8.5 fatal-error-backtraces path, confirmed as code execution against mod_php, and the session upload progress lifecycle, validated against both mod_php and FPM¹¹.

Affected versions and the patch

The fixed releases are PHP 8.5.12, 8.4.27, 8.3.36, and 8.2.35, published on October 21, 2026 at about 14:00 UTC¹. The earliest affected version is disputed in the primary sources, and the report does not resolve it. Calif states that the earliest affected path dates to PHP 5.4.0 but also cites PHP 7.3 or later for one mod_php path¹³. CrowdStrike dates the HashTable ordering behind CVE-2026-109618 to an October 2017 performance change first shipped in PHP 7.3.0, and the serialization and session code behind CVE-2026-109619 to PHP 7.1 and 5.4⁹. For defenders the version debate is moot. The PHP security team published minimal source backports for 7.4, 8.0, and 8.1, and nothing official exists for 5.4 through 7.3, which are end of life¹⁴. Every branch that serves web requests should be treated as affected until patched or confirmed backported by the distribution vendor.

Why exploitation is hard

The CVSS attack complexity rating is High for a reason. A memory-corruption primitive is not code execution. Turning either flaw into a shell on a modern x86-64 host requires defeating address space layout randomization, which a memory leak alone does not provide, and it requires tuning memory pressure to the specific build and prior allocations³. Independent reviewers reached the same conclusion in the first days. VulnCheck found no public proof of concept that clears this bar on x86-64. watchTower's Jake Knott said the exploits being sprayed either do not work, test only for timing differences, or crash the target¹⁵. Justin Elze reviewed the public GitHub proofs of concept and found none that worked against a real target². The one public lab chain that implements a real memory leak is limited to a calibrated PHP 8.4.26 Apache setup with a single prefork worker and target-specific offsets³. This is the hinge of the whole assessment. The bugs are real and sit in the engine, but reliable internet-wide exploitation was not practical as of October 23.

Exploitation activity and the attempt-versus-success gap

Attempt volume is large and success evidence is absent. Amazon detected the first exploitation attempt against its MadPot honeypots on October 17, four days before disclosure, from an IP tied to a commercial anonymization network, and more attempts followed on October 21 between about 02:00 and 08:00 UTC from other anonymized infrastructure¹⁶. Cloudflare reported thousands of attempts in the ten hours before disclosure, tens of thousands within 45 minutes of disclosure, hundreds of thousands within two hours, and millions by early October 22, with attackers mutating payloads to evade the web application firewall¹⁷. Against that, the success column is empty. Amazon says it is unclear whether any attempt would succeed on a real system. VulnCheck's canaries recorded one webshell plant attempt from a Chinese IP on October 21 in which all ten file-read requests returned 404⁴. Hunt.io tracked a mass run that grew to 123 targets and recorded no code execution and no shells. SOCRadar found an attacker had screened about 20 sites with randomization tooling, and none met the exploit's preconditions⁸. As of October 23, no source has named a victim with post-exploitation evidence.

One official statement moved against that pattern. On October 22 at 16:00 the Dutch NCSC said it was investigating initial signs of successful exploitation, a shift from its earlier statement that it saw no sign of active misuse¹⁸. The agency released no detail, and a Dutch press account reported that the observed attempts did not appear to succeed. The statement is a possible success, not a confirmed one, and it is the single most important item to watch.

Exploit capability as of October 23

Private capability is further along than public capability. VulnCheck has a weaponized exploit for CVE-2026-109618, available to its Initial Access Intelligence customers, that chains a memory leak to code execution on x86-64 with randomization enabled and `display_errors` off, the recommended production setting, and covers PHP 8.4.10 through 8.4.26 unmodified⁴. Its ARM64 memory leak works but reliable code execution on ARM64 does not yet. CrowdStrike reports a test exploit that spawns a shell from a `mod_php` worker with a single unauthenticated POST, works on x86-64 and ARM, and defeats randomization on ARM over HTTP alone⁹. The two vendors disagree on ARM. CrowdStrike reports a working ARM randomization bypass over HTTP while VulnCheck reports ARM64 code execution as not yet functional, a discrepancy most likely explained by different test builds, which neither has confirmed. On the criminal side, SOCRadar recovered a lab-built one-click chain named for randomization bypass, calibrated to PHP 8.4.26 under Apache `mod_php`, from a server assessed as financially motivated with a cryptocurrency and Web3 focus⁸. A Mirai variant tracked as Potassium added PHP2Shell to its spreading kit, pulling a loader from a known payload host, and its attempts were unsuccessful¹⁹.

Exposure

Exposure counts vary widely because vendors count different things, and they should not be compared across sources. PHP runs on roughly 69 to 70 percent of websites by one common measure, which counts sites and not exploitable hosts¹⁴. Host and service counts range from about 2.58 million hosts seen by Censys on October 21, of which about 1.57 million reported an unpatched version, up to 3.08 million hosts and 56.7 million web properties in the October

22 Censys update, a rise the firm attributes to better detection rather than growth²⁰. VulnCheck counts about 2.6 million PHP hosts and assesses 99.998 percent as vulnerable or likely vulnerable, with over 1.1 million on end-of-life branches⁴. Censys reported about 12.8 million affected services in the United States and cautioned that version-based counts may overstate exposure²¹. A figure of 34.4 million vulnerable services, attributed by The Hacker News to Censys as of October 23, could not be found on the Censys advisory page and is treated as unverified¹⁹. The consistent finding across all sources is that most exposed PHP runs on end-of-life branches that will get no official fix.

Disclosure failure and the compressed timeline

The coordinated disclosure broke down, and that compressed the defender timeline. CVE identifiers leaked onto forums and social media on October 20 with a wrong score and release time. Calif then mistakenly posted a video showing a proof of concept against live targets, which was deleted but spread, and which the PHP Foundation says sped up exploit development¹. Chainguard published patched container images at 07:41 UTC on October 21, more than six hours before the coordinated release, which handed attackers the fix in binary form for comparison. Chainguard's CEO says the firm independently found the same flaw on October 16 using its own AI tooling, joined the coordination channel, and released early after several vendors reported active exploitation overnight while the European maintainers were asleep²². The PHP Foundation moved the release up a day in response. The net effect for subscribers is that attacker scanning began about 20 minutes after the fix shipped, and the earliest honeypot hit predates the patch by four days¹⁶.

Official response

Government response has been consistent and precautionary. United States agencies urged immediate patching or disconnection of systems that cannot be secured, and CISA said on October 21 that it had not observed exploitation as of publication, while the FBI advised updating or taking affected systems offline²³. Neither CVE appears in the CISA Known Exploited Vulnerabilities catalog according to every secondary source checked on October 23. The catalog itself could not be reached directly from the research environment, so the KEV status is reported as unverified against the primary source²⁴. The United Kingdom NCSC sent about 50,000 warnings and advised patching or taking systems offline. A UK minister told the House of Lords on October 22 that 32 government services were down and four partially down, largely as precaution, with no confirmed compromise of central government or the wider public sector²⁵. The Netherlands, Germany, France, Belgium, Finland, Denmark, and others took services offline as precaution, and most were returning by October 22^{26,27}. In Japan, Mizuho, Shizuoka Bank, MUFG, ANA, JAL, JR East, and Seven-Eleven Japan took services offline, the prime minister urged patching, and the government reported no confirmed attacks²⁸. Canada's Cyber Centre issued alert AL26-026 on October 21 and said it had not identified active exploitation²⁹.

SECTION 5: ANALYSIS

Assumptions

Assumption 1: No portable, default-configuration, x86-64 remote code execution exploit is publicly available as of October 23. This is supported by VulnCheck, watchTowr, and independent reviewers who found no working public proof of concept against real x86-64 targets. It fails the moment a universal public exploit appears, at which point attempt volume converts to compromise volume.

Assumption 2: Attempt telemetry is a weak proxy for success. This is supported by the uniform pattern across Amazon, Cloudflare, Hunt.io, SOCRadar, and VulnCheck of high attempts and no confirmed shells. It fails if vendors are failing to detect successful compromise rather than observing its absence, which the memory-disclosure path in Key Judgment 4 makes plausible.

Assumption 3: End-of-life and embedded PHP will patch slowly or never. This is supported by the exposure data showing roughly 42 percent of hosts on unsupported branches and by the structural fact that embedded PHP follows vendor timelines. It fails only if distributions and appliance vendors backport fixes far faster than history suggests.

Assumption 4: The Dutch NCSC signal remains unconfirmed or resolves as unsuccessful attempts. This is supported by the lack of released detail and by the Dutch press account. It fails if the Netherlands confirms a genuine compromise, which would be the first named post-exploitation evidence and would move Key Judgment 1.

LINCHPIN ASSUMPTION: Assumption 1 is the linchpin. The entire assessment that near-term mass compromise is unlikely rests on the fact that reliable code execution against randomized, default-configured, modern x86-64 systems is not yet a commodity. If a portable public exploit that defeats randomization under production settings is released, the millions of daily attempts already in flight would begin to land, and Key Judgment 1 would invert from unlikely to likely within days. This assumption should be revisited at every production cycle and on any credible report of a working public x86-64 chain.

Analysis of Alternatives

For Key Judgment 1, the primary alternative hypothesis is that reliable mass exploitation is already occurring and is simply undetected, because exploitation runs below the application layer and leaves little in application logs. We assess this alternative as unlikely. Endpoint and honeypot vendors watch process lineage and file-system activity, not application logs, and they report no shells. If mass compromise were underway, the shells would show.

For Key Judgment 2, the primary alternative hypothesis is that end-of-life systems escape compromise because attackers concentrate on high-value modern targets and ignore the long tail. We assess this alternative as unlikely. The recovered criminal tooling is calibrated to old builds and shared hosting, and financially motivated operators favor the weakest systems, not the most valuable.

For Key Judgment 3, the primary alternative hypothesis is the Y2K outcome, that the flaw is remembered as an overreaction with no lasting compromise tail. We assess this as unlikely. Y2K had a hard deadline after which risk vanished. PHP2Shell has a permanent install base of unpatchable systems, so the risk does not expire, it decays slowly.

For Key Judgment 4, the primary alternative hypothesis is that full code execution, not memory disclosure, drives the first real harm. We assess this as roughly even chance. Code execution is the higher-impact outcome and capable actors are close to it, so the ordering of first harm between disclosure and shell is genuinely uncertain.

For Key Judgment 5, the primary alternative hypothesis is that the PHP patch cycle and the hardening of the fix stall broad exploit diffusion before it matters. We assess this as unlikely. Commercial and criminal x86-64 capability already exists, patch adoption is slow, and the unpatchable tail gives any working exploit a durable target set.

Implications by Customer Sector

Financial Services: The Japanese banking shutdowns show the sector will choose availability loss over breach risk, and that choice is defensible for high-value systems. The priority now is to confirm every internet-facing PHP runtime is patched, rotate any secrets reachable during the pre-patch window, and bring services back rather than extend costly outages on systems already fixed. Watch for credential-theft breaches in coming months that may not be labeled as PHP2Shell.

Critical Infrastructure and Government Services: This sector drove the visible shutdown wave and holds the only possible-success signal, the Dutch NCSC statement. The exposure here is concentrated in legacy portals and bundled PHP. Inventory embedded PHP in appliances and control panels, keep unpatchable internet-facing systems isolated until upgraded, and treat the NCSC signal as the trigger to re-examine logs from October 17 forward.

Healthcare: Hospital websites were among the Dutch systems taken offline. Healthcare PHP often sits in older web properties and third-party portals that patch slowly. Prioritize patching and credential rotation on patient-facing and identity systems, and assume memory disclosure may have exposed session tokens during any unpatched window.

Maritime Commerce: The Port of Rotterdam took systems offline as precaution. Port, logistics, and vessel-facing web systems frequently run embedded or legacy PHP on vendor timelines, which places this sector squarely in the long tail of Key Judgment 2. Map every PHP dependency in port community systems and logistics platforms, press vendors for backport status, and isolate anything that cannot be patched.

Technology and Web Hosting Providers: Shared hosting is the single largest concentration of unpatchable, internet-facing PHP, and recovered criminal tooling targets it directly. Hosting providers should patch the platform, force worker restarts, apply request filtering at the edge as a stopgap, and plan customer communications for the long tail. A provider that leaves end-of-life customer sites exposed is the most probable path to a named compromise.

Changes from Prior Assessment

This report establishes the EOTISEC baseline on PHP2Shell. No prior EOTISEC assessment exists on CVE-2026-109618 or CVE-2026-109619. Subsequent updates should track the running count of named victims with post-exploitation evidence, the CISA KEV status once the catalog is reachable, the resolution of the Dutch NCSC signal, and any release of a portable public x86-64 exploit.

SECTION 6: INFORMATION GAPS AND COLLECTION REQUIREMENTS

- 1. Gap:** Whether any real system has been successfully compromised. The Dutch NCSC reports initial signs of success with no detail, and no other source names a victim. **Analytical significance:** Key Judgment 1 is most sensitive to this. A single confirmed compromise of a patched or default modern system would move the primary judgment. **Collection requirement:** Monitor NCSC-NL alerts and its GitHub repository, watchTower and VulnCheck updates, and incident-response disclosures for the first named post-exploitation evidence.
- 2. Gap:** Whether a portable public x86-64 exploit exists. Current public proofs of concept fail without a second weakness or absent randomization. **Analytical significance:** This is the linchpin. Its appearance would invert Key Judgment 1. **Collection requirement:** Track GitHub proof-of-concept repositories, VulnCheck and watchTower exploit reporting, and exploit-broker chatter for a working default-config x86-64 chain.
- 3. Gap:** The CISA KEV status against the primary source. Secondary sources agree neither CVE is listed, but cisa.gov could not be reached from the research environment. **Analytical significance:** A KEV listing would signal confirmed federal observation of exploitation and would support Key Judgment 2. **Collection requirement:** Check the CISA KEV catalog directly from a browser on each cycle.
- 4. Gap:** Patch adoption rate on supported branches and backport coverage for end-of-life branches. Vendors describe adoption as slow but publish no clean curve. **Analytical significance:** Key Judgment 3, the long-tail trajectory, depends on how fast the patchable population closes. **Collection requirement:** Track the Censys patch dashboard and distribution advisories for 7.4, 8.0, and 8.1 backports.
- 5. Gap:** Whether memory disclosure is already being used to steal secrets. The leak path is confirmed to work, but no credential-theft incident is tied to it. **Analytical significance:** Key Judgment 4 rests on inference here. **Collection requirement:** Watch for breach disclosures and credential-abuse campaigns in the coming months that trace back to the pre-patch exposure window.
- 6. Gap:** The true affected-version floor. Primary sources disagree between 5.4.0 and 7.3.0. **Analytical significance:** This sizes the unpatchable long tail in Key Judgment 2. **Collection requirement:** Reconcile the Calif, CrowdStrike, and PHP advisory version claims as the vendors publish deeper root-cause analysis.

SECTION 7: SOURCE SUMMARY STATEMENT

This assessment rests on open-source reporting only. No classified sources were used. The sources fall into five types. Primary developer and advisory sources, the PHP Foundation release post and the two GitHub Security Advisories, which are authoritative for the technical facts and the patch. Government publications from the United States, United Kingdom, Netherlands, Germany, Japan, and Canada, which carry the official positions and the shutdown record. Security-vendor research and telemetry from Calif, Wiz, CrowdStrike, VulnCheck, Amazon, Cloudflare, Hunt.io, SOCRadar, Recorded Future, and watchTowr, which supply the exploitation and exposure data and carry commercial bias. Established security and trade press, including The Register, SecurityWeek, TechTarget, and Nextgov, used mainly to confirm events and official statements. One independent analyst write-up, Berardinelli, and one independent summary site, php2shell.uk, used for corroboration of the exploitation-difficulty analysis.

Several limitations apply. The claim that no successful compromise has occurred is a claim about absence of evidence across many vendors, not proof of absence, and the Dutch NCSC signal is an acknowledged possible exception. The exposure figures are vendor telemetry that counts different units and cannot be reconciled, so they are reported as ranges and not compared across sources. Vendor exploit-capability claims, particularly the ARM disagreement between CrowdStrike and VulnCheck, are first-party and partly unverifiable. The CISA KEV status rests on secondary sources because the primary catalog could not be reached. The earliest-affected-version range is contradicted within Calif's own write-up and between vendors, and is left unresolved. No single Key Judgment rests on a lone source.

SECTION 8: ICD 503 CONSIDERATIONS AND ATT&CK / D3FEND MAPPING

ICD 503 considerations apply because PHP2Shell is a direct threat to the confidentiality, integrity, and availability of internet-facing information systems that process untrusted input. The vulnerabilities allow unauthenticated remote code execution as the web-server user, which places the risk at the system boundary that most authorization documentation treats as the primary control surface.

Risk to Information Systems and Data: The threat affects any system running PHP to serve web requests, with Apache mod_php at highest risk and PHP-FPM, FrankenPHP, and LiteSpeed also exposed through application paths. The memory-disclosure path puts request-scoped secrets at risk across tenants on the same worker, which means passwords, session tokens, API keys, and certificates belonging to other users can leak from a single process. Shared-hosting and multi-tenant platforms carry the widest blast radius.

NIST RMF Alignment: The Monitor and Assess phases are most directly implicated. Organizations should update the system security plan for internet-facing PHP systems to reflect the new unauthenticated RCE exposure, re-run vulnerability assessment against the fixed versions, and confirm continuous monitoring covers process-lineage detection on web servers. The Implement phase is implicated for systems that applied interim configuration mitigations, which must be documented as temporary and not as remediation.

Supply Chain Risk: PHP is embedded in a long list of third-party appliances, control panels, and web applications, so the supply-chain exposure extends well past first-party code. Organizations under DFARS and similar federal supply-chain obligations should treat bundled and embedded PHP in vendor products as in scope for their inventory and should seek vendor backport status as part of supplier risk tracking. The NCSC-NL software list is a useful starting inventory of PHP-dependent products.

ATT&CK Technique Mapping

The techniques below map the observed and assessed behaviors. REPORTED marks behavior with telemetry or forensic evidence in the sources. ASSESSED marks analytical inference from the source material.

Technique ID	Technique Name	Tactic	Application to This Report	Status
T1595.002	Active Scanning: Vulnerability Scanning	Reconnaissance	VulnCheck canaries recorded scanners with php2shell user agents probing hundreds of hosts, and fingerprinting scripts were recovered from attacker servers.	REPORTED
T1190	Exploit Public-Facing Application	Initial Access	Both CVEs are unauthenticated RCE against internet-facing PHP. Attempt traffic is confirmed across many sensors. No successful shell is confirmed.	REPORTED
T1203	Exploitation for Client Execution	Execution	Successful exploitation would run attacker code in the web-worker process. Demonstrated in vendor labs, assessed for the wild.	ASSESSED
T1588.005	Obtain Capabilities: Exploits	Resource Development	Criminal and commercial actors have acquired or built randomization-bypass exploit chains, recovered from an attacker server and sold commercially.	REPORTED
T1587.001	Develop Capabilities: Malware	Resource Development	The Potassium Mirai variant was extended with a PHP2Shell spreading module.	REPORTED
T1105	Ingress Tool Transfer	Command and Control	Observed attempts fetch an ELF or Go loader from an external payload host following the exploit request.	REPORTED
T1552.001	Unsecured Credentials: Credentials in Files	Credential Access	The memory-disclosure path can leak secrets from process memory across requests, enabling credential theft without a shell.	ASSESSED
T1498	Network Denial of Service	Impact	The Potassium loader is built for ICMP, TCP, UDP, and OpenVPN floods against a Ukrainian network.	REPORTED

Kill-chain phases beyond initial access and resource development are weakly sourced because no confirmed post-exploitation activity exists. Persistence, lateral movement, and exfiltration are not mapped, since the source material supports only attempts and staged capability.

D3FEND Countermeasure Mapping

Priority designations below are grounded in the specific threat. HIGH applies where the control addresses the confirmed primary attack path or the unpatchable population. MODERATE applies to stopgaps and secondary paths.

D3FEND Technique	D3FEND ID	Addresses ATT&CK	Application and Priority
Software Update	D3-SU	T1190, T1203	Patch to 8.5.12, 8.4.27, 8.3.36, or 8.2.35 and restart all workers. This is the only complete remediation per the PHP security team. Priority: HIGH.
Application Configuration Hardening	D3-ACH	T1190	Interim only, and incomplete. Set memory_limit=-1 for CVE-2026-109618, and session.upload_progress.enabled=0 plus fatal_error_backtraces=0 on 8.5 for CVE-2026-109619, then restart workers. Priority: MODERATE as a stopgap on systems awaiting patch.
Inbound Traffic Filtering	D3-ITF	T1190	Edge WAF and request filtering on multipart body size, part count, and header count, including requests without Content-Length. Attackers are mutating payloads to evade WAF, so this is a stopgap only. Priority: MODERATE.
Process Lineage Analysis	D3-PLA	T1203, T1105	Alert on web-server or PHP processes spawning sh, bash, cmd, powershell, python, perl, curl, wget, or nc. This is the primary detection for a successful shell. Priority: HIGH.
Process Spawn Analysis	D3-PSA	T1203	Hunt for the log signature of memory exhaustion during request startup, the Allowed memory size exhausted error paired with Unknown on line 0, which indicates no PHP file was executing. Priority: HIGH.

D3FEND Technique	D3FEND ID	Addresses ATT&CK	Application and Priority
Network Traffic Analysis	D3-NTA	T1105, T1498	Block and hunt the payload host and C2 domains in the IOC set, and alert on outbound ELF or Go fetches from PHP processes. Priority: MODERATE.
Credential Rotation	D3-CR	T1552.001	Rotate credentials, API keys, and session secrets reachable by internet-facing PHP during any unpatched window, then invalidate sessions. Priority: HIGH for systems exposed before patching.
System Isolation	D3-SI	T1190	Isolate internet-facing end-of-life PHP that cannot be patched, per CERT-EU and CERT-FR guidance. Priority: HIGH for the unpatchable long tail.

Priority designations are grounded in report-specific threat data and not assigned generically. The two HIGH detection controls reflect that process lineage and the memory-exhaustion log signature are the only reliable evidence of a successful shell, since application logs show little. The HIGH rating on credential rotation and isolation reflects Key Judgments 4 and 2 respectively.

SECTION 9: ANALYTIC TRADECRAFT SELF-CERTIFICATION

ICD 203 Requirement	Status	Notes
Objectivity: The analysis is free of advocacy, personal preference, or policy bias.	CONFIRMED	<i>The report resists both the hysteria and the overreaction narratives, and grounds the unlikely mass-compromise judgment in the exploitation-difficulty evidence rather than in either framing.</i>
Independence of Political Consideration: No judgment was shaped to support a particular outcome or audience preference.	CONFIRMED	<i>The shutdown-cost finding and the long-tail finding cut in opposite directions for different sectors, and neither was softened for subscriber comfort.</i>
Timeliness: Source material is current and the report addresses matters actionable by the customer.	CONFIRMED	<i>All sources date from October 17 to 23, 2026, and the recommendations are actionable now, during the active patch and exposure window.</i>
Based on All Available Sources: All source material provided was considered. Gaps are documented in Section 6.	CONFIRMED	<i>The narrative tracker supplied by the customer was read in full, and every load-bearing claim was re-verified against the primary source before use. Six gaps are documented.</i>
Source Credibility Described: Source quality is addressed in Section 7.	CONFIRMED	<i>Section 7 names five source types, flags vendor commercial bias and the ARM-capability disagreement, and notes the KEV and version-floor limitations.</i>
Uncertainty Expressed: Probability and confidence language conforms to ICD 203 standards.	CONFIRMED	<i>Each judgment carries one ICD 203 probability term and a separate confidence level with rationale, and no sentence combines the two.</i>
Assumptions Distinguished from Facts: All assumptions are labeled. Linchpin assumptions are identified.	CONFIRMED	<i>Four assumptions are numbered, and the linchpin, the absence of a portable public x86-64 exploit, is identified and tied to the primary judgment.</i>
Alternatives Incorporated: At least one alternative hypothesis is addressed for each major judgment.	CONFIRMED	<i>One alternative is assessed for each of the five Key Judgments, each with ICD 203 probability language.</i>
Customer Relevance Addressed: Implications for the relevant stakeholder are stated specifically.	CONFIRMED	<i>Section 5 gives sector-specific implications and actions for all five sectors named in Section 1.</i>
Clear and Logical Argumentation: The main analytic message is stated up front in Section 3.	CONFIRMED	<i>Five Key Judgments lead the report, and the evidence in Section 4 and the analysis in Section 5 support them in order.</i>
ATT&CK and D3FEND Mapping Completed: REPORTED and ASSESSED attributions are correctly distinguished. Priority designations are grounded in the specific incident.	CONFIRMED	<i>Eight ATT&CK techniques are marked REPORTED or ASSESSED by evidence type, and eight D3FEND countermeasures carry priorities grounded in the specific threat.</i>

SECTION 10: ENDNOTES

1. PHP Foundation. "Today's PHP security release and how we're strengthening PHP security." thephp.foundation, October 21, 2026. Page fetched and read.
2. Hunt.io. "PHP2Shell (CVE-2026-109618, CVE-2026-109619): Open Directories and Exploit Tooling." hunt.io, October 22, 2026. Page fetched and read.
3. Berardinelli, Daniele. "Why php2shell can't be widely exploited." berardinellidaniele.com, October 22, 2026. Page fetched and read.
4. VulnCheck. "PHP2Shell Vulnerabilities (CVE-2026-109618, CVE-2026-109619)." vulncheck.com, October 21, 2026, updated October 22, 2026. Page fetched and read.
5. Recorded Future, Insikt Group. "PHP2Shell Exploitation." recordedfuture.com, October 21, 2026. Page fetched and read.
6. BISI. "PHP2Shell: Critical PHP Vulnerabilities and the Fragility of the Web's Foundations." bisi.org.uk, October 23, 2026. Page fetched and read.
7. Wiz Research. "PHP2Shell: AI uncovers critical RCE vulnerabilities affecting 70% of the Internet." wiz.io, October 21, 2026. Page fetched and read.
8. SOCRadar. "PHP2Shell Exploit Tooling Found on Attacker Server." socradar.io, October 22, 2026. Page fetched and read.
9. CrowdStrike. "CrowdStrike Protects Customers from PHP RCE Zero Days." crowdstrike.com, October 21, 2026, updated October 22, 2026. Page fetched and read.
10. PHP. "RCE via OOM corruption." GitHub Security Advisory GHSA-w8hw-w9fx-j79w, CVE-2026-109618, October 21, 2026. Page fetched and read.
11. PHP. "RCE via stale global state." GitHub Security Advisory GHSA-6wmf-4rv5-p3rp, CVE-2026-109619, October 21, 2026. Page fetched and read.
12. Daubois, Alexandre. "CVE-2026-109618 php2shell rumors." daubois.dev, October 21, 2026. Page fetched and read.
13. Calif (Ryan Austin). "PHP2Shell." calif.io, October 2026. Page fetched and read.
14. Tenable. "PHP2Shell CVE-2026-109618, CVE-2026-109619: Frequently Asked Questions." tenable.com, October 2026. Page fetched and read.
15. SecurityWeek. "PHP2Shell Exploitation Started Days Before Patches." securityweek.com, October 22, 2026. Page fetched and read.
16. Amazon Web Services. "Amazon threat intelligence identifies PHP2Shell exploitation prior to public disclosure in MadPot." aws.amazon.com, October 21, 2026, last modified October 22, 2026. Page fetched and read.
17. The Register (Carly Page). "Hackers pile on PHP2Shell as AWS reveals attacks began before disclosure." theregister.com, October 22, 2026. Page fetched and read.
18. NCSC-NL. "Update: PHP vulnerabilities remain a reason for immediate patching." ncsc.nl, published October 21, 2026, updated October 23, 2026 12:15. Page fetched and read.
19. The Hacker News (Ravie Lakshmanan). "PHP2Shell Exploit Attempts Hit Millions as Attackers Try to Deploy Mirai Malware." thehackernews.com, October 23, 2026. Page fetched and read.
20. Censys. "CVE-2026-109618 and CVE-2026-109619 Advisory." censys.com, published October 21, 2026, updated October 22, 2026. Page fetched and read.
21. Nextgov/FCW. "PHP flaws draw millions of hacking attempts amid global push for fixes." nextgov.com, October 22, 2026. Page fetched and read.
22. TechTarget. "PHP2Shell disclosure unravels after leaks forced early patch drop." techtarget.com, October 22, 2026. Page fetched and read.
23. ABC News wire (via 6abc). "Urgent security fix issued for code vital to estimated 70% of all websites." 6abc.com, October 21, 2026. Page fetched and read.
24. php2shell.uk. "PHP2Shell FAQ." Independent vendor-neutral summary, reviewed October 23, 2026. Page fetched and read.
25. The Register (Carly Page). "PHP2Shell puts dozens of UK government services out of service." theregister.com, October 22, 2026. Page fetched and read.
26. Reuters (Raphael Satter and others), via The Star. "EU agency warns websites may need to be taken offline as PHP prepares patches." thestar.com.my, October 21, 2026. Page fetched and read.
27. TechRadar Pro live blog. "PHP issues affect websites across the world." techradar.com, October 21 to 22, 2026. Page fetched and read.

28. Coruzant Technologies. "PHP2Shell vulnerability draws millions of attacks." coruzant.com, October 22, 2026. Page fetched and read.
29. Canadian Centre for Cyber Security. "AL26-026: Critical vulnerabilities impacting PHP." cyber.gc.ca, October 21, 2026. Page fetched and read.