

FOUR INCIDENTS · ONE WEEK · OCTOBER 2026

BLACK OCTOBER

A PHP engine flaw, a certificate authority whose keys anyone could compute, a CVSS 10 in GitLab and a hijacked AI-gateway release. What was confirmed, what was only claimed — and what to do now.

PRIVATE KEY, EM OV TLS CA - G3C

4

One intermediate CA's private key was the integer four. Others were 56 and 115.

Page 7.

04

PHP

PHP2Shell: the bug beneath the application — versions, SAPIs and what to patch first

07

CA

eMudhra: eight CA keys anyone could compute — and the clients browsers can't protect

09

GITLAB

A real CVSS 10, no evidence of attacks — yet — hunting on a host that may leave no trace

11

LITELLM

A five-month-old key, a trusted pipeline — inside the 11070 stealer

Every claim graded:

CONFIRMED

REPORTED

ATTEMPTED

UNVERIFIED

Evidence reviewed through 25 Oct 2026

Also inside: Blue-team field lab · What leaders actually said · 24-hour playbook · Sources & fact-check log

No AI-generated imagery

THE EDITORIAL DESK

The new speed of risk

Four incidents in five days — a language-runtime RCE, a certificate-authority key failure, a CVSS 10 in a DevSecOps platform, and a hijacked AI-gateway release pipeline. This edition separates what was confirmed from what was claimed, and gives defenders the versions, indicators and commands they need.

EDITORIAL PERSPECTIVE

Artificial intelligence has changed the speed at which vulnerabilities are discovered and weaponised. The answer is not to fear AI but to become more disciplined: know every exposed asset, verify patches rather than assume them, monitor continuously, and use AI defensively under human oversight. A strong security culture is still our greatest advantage.

El Mostafa Ouchen · Founder & Editor, MAG212 CYBER · 25 October 2026

HOW TO READ THIS EDITION

CONFIRMED

Stated by the vendor, root program or project in a primary advisory.

REPORTED

Published by a named security firm or outlet; not yet confirmed by the vendor.

ATTEMPTED

Exploit traffic observed. Attempts do not establish compromise.

UNVERIFIED

Circulating but unsupported by any primary source. Treat as rumour.

INSIDE THIS ISSUE

03	THE WEEK	Black October timeline
04	PHP2SHELL I	Discovery, CVEs, scoring
05	PHP2SHELL II	Versions, SAPIs, distros
06	PHP2SHELL III	Exploitation, IOCs, mitigations
07	TRUST	eMudhra: keys anyone could compute
08	TRUST II	Root-program response & exposure
09	DEVSECOPS	GitLab CVE-2026-110264
10	DEVSECOPS II	Detection and hunting
11	SUPPLY CHAIN	LiteLLM 1.107.0 attack chain
12	SUPPLY CHAIN II	Payload, IOCs, response
13	FIELD LAB	Safe triage commands
14	LEADERSHIP	What leaders actually said
15	STRATEGY	Governed AI defence & 24-hour plan
16	INTEGRITY	Fact-check log & sources
17	COLOPHON	Publisher, disclosures, licence

MAG 212 CYBER

Produced and published by MAG212 CYBER · cyber.mag212.com. Technical analysis, field commands and response guidance in this edition were prepared for IT teams and security leaders.

THE WEEK IN NUMBERS

9.2

CVSS v4.0 for both PHP2Shell CVEs (pre-auth RCE demonstrated)

10.0

CVSS 3.1 for GitLab CVE-2026-110264 (unauthenticated RCE)

8

eMudhra intermediate CA keys — one of them the integer 4

~49k

Est. downloads of backdoored litellm 1.107.0 (Wiz, single source)

COMMON THREAD

In every case **the exposure outlasts the patch**: end-of-life PHP branches, CA keys that are now public forever, GitLab hosts that stopped at the 12 October release, and LiteLLM hosts still polling an attacker dead drop.

CHRONOLOGY

Black October, hour by hour

Four independent incidents overlapped in one week. All times UTC. PHP CA GITLAB LITELLM

- 27 MAY

LITELLM Agent Platform master key committed in `opencode.json` to public repo `LiteLLM-Labs/lite-harness`. Later force-pushed over — **never revoked**.
- 19 AUG

PHP First PHP engine bug found (Wiz, AI-assisted). Reported to PHP 24 Aug; acknowledged 21 Sep.
- 10 OCT

LITELLM 21:13–21:28 — attacker uses leaked key against `/api/v1/managed_agents/agents`, obtains an org owner's GitHub CLI token; CI secrets exfiltrated to `webhook.site`.
- 12 OCT

GITLAB Release 19.4.2 fixes a *different* CVSS 10 regex flaw (CVE-2026-97229). LITELLM Second secret-theft round across private repos.
- 13 OCT

LITELLM Exfil domain `litellm.dev` registered (00:45).
- 17 OCT

PHP Amazon MadPot honeypot records a PHP2Shell attempt from 103.216.220[.]92 — **four days before disclosure**.
CA 18:05 Google, 20:06 Mozilla receive the eMudhra weak-key report. Chrome CRLSet block 19:45; Firefox OneCRL block live by 22:59.
- 18–19 OCT

CA eMudhra revokes its 8 intermediates (18 Oct); Sectigo revokes the cross-signs (19 Oct 10:35). First post-disclosure rogue certificate hits CT at 20:24 on 19 Oct.
- 20 OCT

PHP CVE IDs leak on Hacker News; php.net posts advance notice. Release pulled forward a day.
CA Mozilla announces publicly; emSign ECC Root CA - G3 removed in NSS 3.131.1.
- 21 OCT

PHP 03:02 Cloudflare WAF rule live; 14:00 embargo lifts; PHP 8.5.12 / 8.4.27 / 8.3.36 / 8.2.35 ship.
GITLAB Critical Patch Release 19.5.1 et al. (CVE-2026-110264).
LITELLM 11:39 malicious 1.107.0 wheels on PyPI; 11:46 containers; quarantined 13:23 (PyPI), 14:20 (Docker Hub).
- 22 OCT

GITLAB CVE record public (00:33); CERT-EU advisory 2026-019. CA eMudhra exits global SSL/TLS business.
- 23 OCT

PHP PHP publishes informational CVE-2026-115560.
LITELLM Canonical attack-chain post published.
- 27 OCT

CA Chrome Root Store v43 (root removal) expected in Stable.

FOUR FACTS, FOUR FIELDS

Record each separately in every incident ticket. Conflating them is the most common reporting error of the week.

Exposure	The vulnerable component is reachable.
Attempt	Exploit traffic was observed.
Compromise	Corroborated evidence of attacker control.
Remediation	Fix verified — not inferred from a version string.

STATUS AT PRESS TIME

INCIDENT	ATTEMPTS	COMPROMISE
PHP2Shell	WIDESPREAD	None public
eMudhra	107K TEST CERTS	None malicious
GitLab	NONE REPORTED	None
LiteLLM	CONFIRMED	malicious release

RE-CHECK BEFORE ACTING

CISA KEV status for the PHP and GitLab CVEs was last confirmed 22–23 Oct (neither listed). Check the live catalog.

RUNTIME SECURITY · PART I

PHP2Shell: the bug beneath the application

Two memory-safety flaws in the PHP engine itself give unauthenticated attackers a path to code execution — before a single line of WordPress, Drupal or Joomla runs.

Who found it

The flaws were disclosed on 21 October by researchers from **Calif (LCFR)**, **Wiz**, **OpenAI** and **Quarkslab**; Wiz's write-up coined the name "PHP2Shell". Wiz demonstrated pre-authentication RCE against default WordPress, Drupal and Joomla container images, plus remote memory disclosure.

The timeline shows how fast AI-assisted research now moves: first bug on 19 August, unauthenticated RCE on Drupal the next day and WordPress the day after, reported to PHP on 24 August, patched 21 October. Wiz credits AI tooling across discovery, exploit development, detection and patch validation. OpenAI says it led a coalition that worked the flaws for two months and released a Codex checker skill.

Why it matters

The vulnerable code dates back to **PHP 5.4.0** (2012). Updating a CMS plugin does nothing — the fault sits in the runtime every PHP application shares. VulnCheck counts ~2.6 million internet-facing PHP hosts; Modat counts 4.38 million services, ~79% on unsupported branches.

THE THREE CVES

CVE / ADVISORY	ROOT CAUSE	CONFIRMED PATHS	FIX
CVE-2026-109618 GHSA-w8hw-w9fx-j79w 9.2	An out-of-memory bailout can interrupt PHP while request data is only partly initialised; request cleanup then treats the invalid state as valid.	OOM <code>longjmp()</code> during multipart parsing → unauthenticated RCE as the web-worker user (validated on default <code>mod_php</code>).	"Soft OOM" at the next VM safe point; hard OOM exits via <code>_exit()</code> .
CVE-2026-109619 GHSA-6wmf-4rv5-p3rp 9.2	Request-owned global state survives after its memory is freed; a later request on the same worker uses the stale reference (use-after-free).	PHP 8.5 fatal-error backtraces (<code>mod_php</code>); session upload progress (<code>mod_php</code> and <code>FPM</code>).	Global state reset at request boundaries.
CVE-2026-115560 23 Oct · informational NO CVSS YET	Session upload progress when an app switches <code>session.serialize_handler</code> to <code>php</code> at runtime before <code>session_start()</code> .	PHP calls it an unlikely RCE vector. Affects fully patched releases, including 8.5.12.	Next regular release.

Note: Wiz's blog maps individual bugs to CVEs differently from PHP's advisories. This edition follows PHP's mapping, which is authoritative. Sources: PHP GHSA advisories; php.net 2026 archive; Wiz; Cloudflare; Tenable; The Hacker News.

SCORING (PHP ADVISORIES)

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

9.2

Critical — both CVEs

CWE-416 · 755
824 · 1325 · 1419

Shared CWE set

A 10.0 score circulated in the pre-release leak; PHP's final rating is 9.2. **AC:H/AT:P** reflects real preconditions (OOM timing, worker reuse, specific features) — not low practical risk.

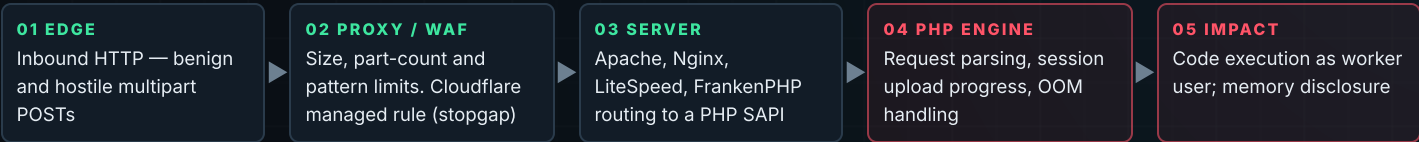
FACT-CHECK

The 23 October session-upload-progress issue was published by the **PHP project** as CVE-2026-115560 — not by Cloudflare, which updated its existing 21 October post.

RUNTIME SECURITY · PART II

Where PHP2Shell lands

An original architecture view — not an exploit recipe. The fault is reached through ordinary request parsing, below every framework control.



AFFECTED AND FIXED VERSIONS

BRANCH	FIXED IN	STATUS
8.5	8.5.12	SUPPORTED
8.4	8.4.27	SUPPORTED
8.3	8.3.36	SECURITY ONLY
8.2	8.2.35	EOL 31 DEC 2026
8.6	8.6.0 RC4	PRE-RELEASE
7.4 / 8.0 / 8.1	Source-only backports	NO OFFICIAL RELEASE
5.4 – 7.3	None	EOL · VULNERABLE

SERVER APIS (SAPIS) IN SCOPE

SAPI	RISK PER PHP PROJECT
Apache mod_php	HIGHEST demonstrated pre-auth exploit before app code runs
PHP-FPM	LIKELY SUSCEPTIBLE session-upload path confirmed
php-cgi, FrankenPHP, LiteSpeed	IN SCOPE
CLI	Not stated. Bugs need a reused web worker — but do not assume CLI is safe.

Vendor impact: the Dutch NCSC tracks affected products at github.com/NCSC-NL/php2shell. Palo Alto lists PAN-OS and Panorama as affected with no fix at press time.

DISTRIBUTION BACKPORTS

DISTRIBUTION	ADVISORY	NOTES
Ubuntu 26.10 / 26.04 / 24.04 / 22.04	USN-8948-1, USN-8948-2 (Pro)	Patched packages shipped on disclosure day
Debian 13 trixie	DSA-6574-1	8.4.27-1~deb13u1
Debian 12 bookworm	DLA-4848-1	8.2.35-1~deb12u1; nothing listed for bullseye
Red Hat Enterprise Linux	RHSB-2026-013 · RHSA-2026:87072, :87080	See bulletin for per-stream packages
Amazon Linux 2023	ALAS2023-2026-3301 ... 3304	AL2 core php and AL2023 php8.1: no fix
CloudLinux · Remi · Chainguard	Vendor channels	Day-one fixes reported

VERSION STRINGS LIE

Distribution backports keep the **old upstream number** in `php -v`. Scanners that compare version strings will flag patched hosts and may pass unpatched custom builds. Verify the package changelog or advisory ID, not the banner. Distro package versions above are from a secondary aggregator — confirm against the primary advisory.

RUNTIME SECURITY · PART III

Attempts, indicators, interim controls

Exploitation attempts are widespread. No confirmed compromise had been documented publicly at press time — keep those two facts apart.

What has been observed

- **Pre-disclosure:** Amazon's MadPot saw one attempt on 17 Oct from 103.216.220[.]92 (anonymisation network). Amazon says it is unclear how the actor obtained exploit details and has not said whether the attempt succeeded.
- **Post-disclosure:** watchTowr reports in-the-wild exploitation attempts since 21 Oct; Cloudflare's CEO cited "millions" of attempts within hours; Pantheon blocked 60,000+ suspicious requests on day one; Singapore CSA (AL-2026-151) reports multiple public PoCs.
- **At release** the PHP project found no functional public exploit. CISA had not listed either CVE in KEV as of 23 Oct; its 21 Oct alert advises considering taking unpatchable or ≤ 8.1 systems offline. CERT-EU advisory 2026-018 recommends isolation.

Indicators (investigate, don't blanket-block)

SOURCE	INDICATOR
AWS · 17 Oct	103.216.220[.]92
AWS · 21 Oct	31.223.184[.]166 · 37.19.200[.]15 · 134.209.159[.]64 · 103.114.163[.]188 · 95.217.98[.]37
Telekom honeypot	45.198.224[.]169 (serialized-data POSTs) · 64.176.35[.]212 (parser fuzzing)

Indicators are defanged ([.]); a machine-readable CSV is attached to this PDF. These are shared anonymisation and cloud addresses. "php2shell-*" User-Agents identify scanners, not attackers.

Interim hardening while you patch

```
; PHP-FPM pool (or php.ini) — NOT ini_set() or .user.ini. Restart all workers.
php_admin_value[memory_limit] = -1          ; enforce memory with cgroups instead
php_admin_value[post_max_size] = 8M         ; 32M if uploads are needed
php_admin_flag[session.upload_progress.enabled] = off
php_admin_value[max_input_nesting_level] = 32
php_admin_flag[fatal_error_backtraces] = off ; PHP 8.5 only
pm.max_requests = 1                        ; only if performance allows
```

Detection signals

- `PHP_SESSION_UPLOAD_PROGRESS` field in multipart POST bodies (AWS)
- Serialized PHP data inside a form parameter — strongest single signal (Telekom)
- Multipart requests followed by "Allowed memory size exhausted" errors (Wiz)
- Bursts of HTTP 500, oversized or malformed `PHPSESSID` cookies, unexplained worker restarts (Wiz)

CLOUDFLARE MANAGED RULE

"PHP Remote Code Execution Attack (CVE-2026-109618, CVE-2026-109619)" — on by default on every plan including Free, live 03:02 UTC 21 Oct.

Free Managed Ruleset	4e315666e8e344cdb64119a3b5ccf32f
----------------------	----------------------------------

Managed Ruleset	c1d409ad838f47f59446f21cc438d5dd
-----------------	----------------------------------

Confirm block mode on paid plans. Sites that renamed `session.upload_progress.name` are not covered. Cloudflare calls it a stopgap.

- Use container/cgroup memory limits — **never** `RLIMIT_AS` or `RLIMIT_DATA`.
- At the proxy, cap request size, field count, multipart parts, header count and nesting; block multipart where unused.
- Disabling upload progress blocks the confirmed session paths, not every variant. Keep it off after patching if unneeded.

Only updating is a complete fix. Source: php.net; PHP GHSAs.

TRUST INFRASTRUCTURE · PART I

eMudhra: eight CA keys anyone could compute

Nobody stole these keys. A flawed key-generation ceremony made them so small that they could be recovered from the public certificates in under a second.

What went wrong

On **30 April 2025** an eMudhra key ceremony produced P-384 private keys whose values were all **below 256**. Researchers recovered keys equal to 4, 56 and 115; BleepingComputer confirmed that the key for "EM OV TLS CA - G3C" is literally the integer **4**.

eMudhra's explanation: a key wrap/unwrap step on its Utimaco HSMs created a second, weak key pair, and similar labels led to that pair being used for the CSRs — including those sent to Sectigo for cross-signing. eMudhra says its RSA hierarchy is unaffected. Exposure window: ~18 months.

How it was found

A researcher identified only as "Wayne" ran Google's open-source **paranoid_crypto** library across CA certificates listed in the CCADB, then proved control on 17 October by signing a test message with each key. Tracked in Mozilla Bugzilla **2083449**.

TRUST CHAIN (SIMPLIFIED)

ROOT

emSign ECC Root CA - G3

REMOVED

ISSUING ROOTS

emSign Root TLS CA - G3 · emSign Root CS CA - G3

8 INTERMEDIATES (WEAK KEYS)

6 TLS (DV/OV/EV × G3B/G3C) · 2 code-signing (OV CS G3B/G3C)

+ 8 CROSS-SIGNS

Sectigo "AAA Certificate Services" — same keys → 16 certificates

CORRECTION

The intermediates were never "in" Mozilla's trust store. Mozilla **blocked** them via OneCRL and **removed the parent root** in NSS 3.131.1. Google received the report first (18:05 UTC); Mozilla announced it on 20 October.

AFFECTED INTERMEDIATES — SPKI SHA-256 (IDENTIFIES BOTH EMUDHRA AND SECTIGO-ISSUED CERTS)

INTERMEDIATE	TYPE	SPKI SHA-256
EM DV TLS CA - G3B	TLS	EBED6475C2C2D80FCB199F3FE8DD6E6B3CE60B77EB15D4E42C3DEC0C6F5B5105
EM OV TLS CA - G3B	TLS	43BFFA19D6EF651451F6841E019931E96F6F2EAEFF28AB4B6F1CE4576DE2717B
EM EV TLS CA - G3B	TLS	E0254C7FDCD265E2F0F8A428DABB75AB4AE039B4872D62F5EE57F0877D5A8A41
EM DV TLS CA - G3C	TLS	49DD353DC2B919A3770645117D0E60C16115E7D1BC259EEFE734B00EA434FE5F
EM OV TLS CA - G3C	TLS	63ECEFEFE431D49523BC1136C4E72D0206A477086CAF606CD5CF025686C109CF8
EM EV TLS CA - G3C	TLS	5C0660A2B164910D8FF589C18E14552E9E0EC237DCEFE2CB736D41A4811F81A0
EM OV CS CA - G3B	Code signing	EC4FFA1AD0AC88081314B7AAFFFEF6A91F97CD57291BB5BA686C5415F5EB4F6B6
EM OV CS CA - G3C	Code signing	92C063D67E8F14DB71D288E11266FFDAF69CBF115F7BAB14F3BC2F68D25E0E30

Source: researcher-maintained impacted-intermediates.json (emudhra.we-have-a-problem.com), prefixes corroborated by ctlogs.dev. All 16 revoked with reason ca_compromise — eMudhra 18 Oct 09:24–09:38 UTC, Sectigo 19 Oct 10:35 UTC. Copy hashes from the source JSON for automated tooling.

TRUST INFRASTRUCTURE · PART II

Browsers are covered. Your other clients may not be.

Root programs blocked the certificates within hours. The keys, however, are public forever — so anything that ships its own CA bundle and ignores revocation is still exposed.

HOW THE ROOT PROGRAMS RESPONDED

PROGRAM	MECHANISM	TIMING AND NOTES
Mozilla	OneCRL block of intermediates; emergency removal of emSign ECC Root CA - G3 (NSS 3.131.1), no distrust-after date	OneCRL live by 22:59 UTC 17 Oct, no restart needed. Firefox 158.0.1, ESR 115.43.1 / 153.5.1 on 20 Oct; Android 158.0.2 on 21 Oct.
Google Chrome	CRLSet block keyed on SPKI (covers Sectigo cross-signs); root removal in Chrome Root Store v43	CRLSet 19:45 UTC 17 Oct. Android via Play Services ~22:00 UTC 19 Oct. v43 expected in Stable 27 Oct. Reviewing eMudhra's other roots.
Microsoft	Automatic CTL update (Disallowed store)	From 20 Oct (per researcher site; no Microsoft advisory found).
Apple	Blocked	Date unconfirmed (sources say 19 or 22 Oct). macOS <code>/etc/ssl/cert.pem</code> update pending — affects <code>/usr/bin/curl</code> , <code>pip</code> .

Certificate Transparency after disclosure

Google says no TLS certificates chaining to the intermediates were logged before the blocks, and eMudhra says none issued end-entity certificates. After disclosure, anyone could sign: the first rogue certificate hit CT at 20:24 UTC on 19 Oct, and by 25 Oct ctlogs.dev counted **107,181** — about 105,900 for google.com. None is classified as malicious.

Code-signing certificates are not logged to CT, so misuse of the two CS intermediates cannot be ruled out that way.

Business impact

eMudhra estimates ~30,000 certificates need migrating to its G1 RSA hierarchy and, on 22 Oct, announced it is exiting the global SSL/TLS business. Site operators chaining to emSign ECC Root CA - G3 must replace certificates before Chrome v43 lands.

Patched trust stores

NSS	3.131.1
Python certifi	2026.10.19
Go	1.27.3 / 1.26.10 (CVE-2026-111943)
Node.js	26.12.1 / 24.22.1 / 22.23.4
Python	3.15.1 / 3.14.9 / 3.13.17
rustls-webpki · webpki-roots	0.103.16 · 1.0.10
Docker Engine · pip · uv	29.9.2 · 26.2.2 · 0.13.5
Alpine ca-certificates	20261020-r0
Distro advisories	DSA-6572-1 · USN-8944-1/-2 · RHSB-2026-012 · CERT-In CIAD-2026-0050

STILL EXPOSED

- Language runtimes and apps with vendored CA bundles
- Container images built before 19 Oct
- Embedded/IoT devices and appliances
- Electron apps; anything not checking CT or revocation

TEST A CLIENT

```
# A certificate ERROR is the good result
curl -sS https://dv-g3b-emudhra.we-have-a-problem.com/ \
  && echo "TRUSTED - update CA bundle" \
  || echo "Rejected (good)"
```

The site's "-noct-" hosts test clients that don't enforce CT. Removing roots also breaks legitimate certificates beneath them.

DEVSECOPS ALERT · PART I

GitLab: a real CVSS 10, no evidence of attacks — yet

An integer overflow in GitLab's bundled regex engine can give an unauthenticated user code execution. Self-managed instances hold source, CI secrets and runner trust — treat this as an emergency upgrade.

The 21 October Critical Patch Release

GitLab shipped **19.5.1, 19.4.3, 19.3.6, 19.2.9, 19.1.9, 19.0.10 and 18.11.13** (CE and EE) on 21 October. The CVE record went public at 00:33 UTC on 22 October, and CERT-EU issued advisory 2026-019 at 09:14 that day — the source of the "22 October" date in early coverage.

The headline flaw, **CVE-2026-110264**, sits in **Onigmo**, the Ruby regex engine GitLab bundles and patches. It affects every release from **7.0** up to the fixed versions. GitLab says exploitation requires "certain conditions" it has not disclosed. Independent analysis (lilting.ch, not confirmed by GitLab) traces the path to the `org-ruby` renderer: attacker-controlled keyword values in an Org-mode (`.org`) file reach Onigmo unescaped, and a public project is enough. Code would run as the `git` service account. Beazley calls it the fourth regex-handling code-execution flaw in GitLab in about a month. Reporter: "joaxcar" via HackerOne.

CVE-2026-110264

10.0

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H

Integer overflow in regular-expression compiler → unauthenticated arbitrary code execution.

GitLab.com Already patched

Dedicated No action on 19.3.5-internal0 / 19.4.2-internal0+

≤ 18.10 NO FIX upgrade line

ALL 12 CVES IN THE RELEASE

CVE	SEVERITY	ISSUE	FIXED IN
CVE-2026-110264	CRITICAL 10.0	Onigmo regex integer overflow → unauthenticated RCE	All seven releases
CVE-2026-15723	HIGH 8.2	Improper authorization in OmniAuth→LDAP account linking (account takeover; requires login)	19.3.6 / 19.4.3 / 19.5.1 only
CVE-2026-112212	HIGH 7.5	OAuth/OIDC discovery cache poisoning	19.3.6 / 19.4.3 / 19.5.1 only
9 further CVEs	5 Medium · 4 Low	See GitLab release post	19.3.6 / 19.4.3 / 19.5.1 only

DON'T STOP AT 12 OCTOBER

The 12 Oct release (19.4.2 / 19.3.5 / 19.2.8) fixed a *different* CVSS 10 — CVE-2026-97229 in the CI/CD regex compiler. Instances patched only on 12 Oct **remain vulnerable** to CVE-2026-110264. Older backport lines (≤19.2) carry only the critical fix.

FACT-CHECK

Claims of in-the-wild exploitation on 21 Oct are **unsupported by any primary source**. GitLab, CERT-EU, Beazley, SOC Prime and Censys reported no confirmed exploitation and no KEV listing as of 22–23 Oct. Unverified "PoC" repositories appeared 22–24 Oct.

DEVSECOPS ALERT · PART II

Hunting on a GitLab host that may leave no trace

GitLab released two detection rules and no IOCs. A successful exploit may leave nothing in the logs — so look at what the attacker would do next, too.

Why GitLab is a high-value target

A self-managed instance concentrates source code, merge history, CI/CD variables, package registries and the trust relationship with every runner. Code execution as `git` can lead to secret theft, malicious pipeline runs, tampered build artifacts and movement into Kubernetes clusters and cloud accounts — a software supply-chain incident.

Urgency is justified by history, not by this CVE: **CVE-2026-85706** (path traversal) was probed the day after its 10 Sep patch and added to KEV on 11 Sep; **CVE-2026-19478** was reportedly exploited within 24 hours in August.

Interim controls (no official workaround)

- Restrict network access to trusted IP ranges / VPN.
- Disable public sign-up; limit public and internal projects.
- These narrow exposure — they do not fix the bug.
- Single-node upgrades incur downtime while migrations run; schedule now.

Post-exploitation hunt list

HOST

- Child processes of Puma or Sidekiq
- Shells or network tools under `git`
- New files in public projects, esp. `.org`

PLATFORM

- New PATs, deploy tokens, OAuth apps
- Unexpected CI jobs, runner registrations
- Admin-setting and audit-event changes

DOWNSTREAM

- Use of GitLab-held secrets elsewhere
- Unexpected registry pushes
- Cloud API calls from runner identities

IF COMPROMISE IS SUSPECTED

Isolate the instance, preserve logs and memory, rotate runner registration tokens, PATs, deploy keys and every CI/CD variable, and validate repository and artifact integrity before trusting the next build.

GITLAB TLDR DETECTION RULES

RULE	SIGNAL (OMNIBUS PATHS)
<code>org_mode_render_error_burst</code>	<code>Timeout::Error</code> rendering <code>.org</code> files, ≥5 from one client in 30 min — <code>/var/log/gitlab/gitlab-rails/exceptions_json.log</code>
<code>onigmo_native_worker_crash</code>	Ruby [BUG] crash reports mentioning Onigmo or org-ruby — <code>/var/log/gitlab/puma/puma_stderr.log</code> (often not shipped to SIEM)

Rule names: CERT-EU. Log mapping: third-party analysis (lilting.ch, ThreatWire).

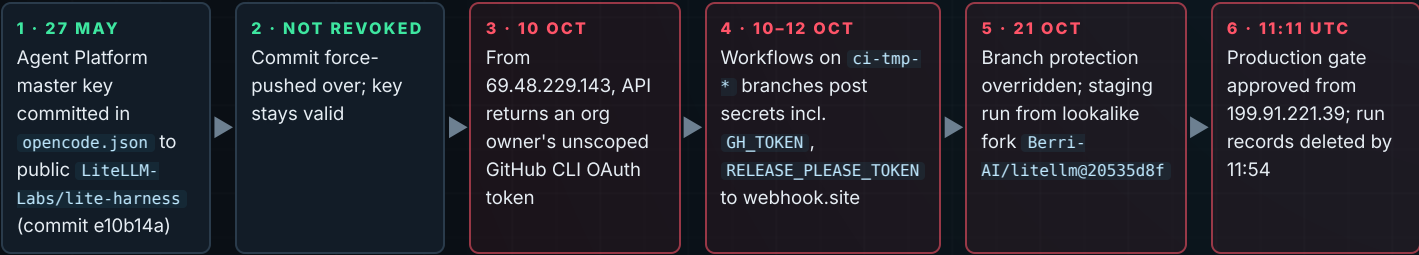
```
# Version check
sudo cat /opt/gitlab/embedded/service/gitlab-rails/VERSION
# Org-mode render timeouts
sudo grep 'Timeout::Error' /var/log/gitlab/gitlab-rails/exceptions_json.log \
| grep -c '\.org'
# Native crashes in the regex engine
sudo grep -iE '\[BUG\].*(onigmo|org-ruby)' /var/log/gitlab/puma/puma_stderr.log
```

SUPPLY-CHAIN FORENSICS · PART I

LiteLLM 1.107.0: a five-month-old key, a trusted pipeline

The attacker didn't steal publishing credentials. They used maintainers' own tokens to make LiteLLM's real release pipeline build and ship a backdoor.

THE VERIFIED ATTACK CHAIN (PER LITELLM'S 23 OCT CANONICAL POST)



Why the controls failed

- **Leaked ≠ revoked.** Force-pushing history does not invalidate a secret.
- **Over-privileged service.** An internal alpha platform stored an owner's unscoped token.
- **CLI tokens skip step-up.** GitHub CLI OAuth tokens are not subject to GitHub's 2FA re-authentication for sensitive actions.
- **Identity looked legitimate.** Three maintainers' tokens were abused; every action was attributed to a real account (Socket).
- **Weakened gate.** Any two engineers could trigger and approve a release.

Exposure windows (21 Oct, UTC)

CHANNEL	LIVE	REMOVED
PyPI · 7 wheels	11:39:51-11:40:13	Quarantined 13:23
GHCR images	11:46:56	13:51
Docker Hub	11:46:56	Repointed 14:20

Tags: `v1.107.0`, `latest`, `main-latest`, `main-stable`. Images: `ghcr.io/berriai/litellm`, `-database`, `-non_root`, `docker.io/litellm/litellm`. Wiz gives slightly earlier removal times. The git tag `v1.107.0`, the `sdist` payload, `rc1/rc2` and LiteLLM's enterprise registry are clean.

Detection: Socket alerted ~7 min after publication; Upwind filed issue #47826. Wiz estimates ~49,000 downloads (single source).

CORRECTIONS

The key was **not** in `BerriAI/litellm` or an `.env.example`. No source says PyPI or registry credentials were stolen. The incident date is **21 Oct** (23 Oct is the write-up). Docker Hub namespace is `litellm/litellm`, not `berriai`.

NOT THE MARCH INCIDENT

In March, TeamPCP published 1.82.7/1.82.8 via a PyPI token exposed in the Trivy compromise (`litellm_init.pth` ; ~47,000 downloads; ~2,500 victim orgs reported). **No link** between October and TeamPCP has been established.

SUPPLY-CHAIN CONTROLS FOR AI GATEWAYS

Secrets

Revoke on leak, then clean history. Scan public org repos (incl. side projects) with push protection.

Identity

Fine-grained, short-lived tokens; no owner tokens in internal services; hardware-key step-up for releases.

Release

Protected environments, required reviewers outside the triggering pair, PyPI Trusted Publishing with attestations.

Consumers

Pin by hash and image digest; delay adoption of brand-new releases; egress allow-lists on gateway hosts.

SUPPLY-CHAIN FORENSICS · PART II

Inside the payload — and getting it out

An AI gateway holds every provider key you own. This stealer was built to take them — and it runs every time Python starts.

COOK.PTH · 12 BYTES

Executes `import cook` at **every** Python start — no import of `litellm` needed. Running `pip` triggers it.

COOK.PY · "PRIVCOLLECT"

Harvests cloud/IMDS, Kubernetes, SSH/GPG, registry, CI and ~20 AI-coding-tool credentials, browser data, `.env`, `/proc/*/environ`; dumps LiteLLM DB key tables if `DATABASE_URL` set. AES-256-CBC + RSA-4096, POST over plain HTTP.

SITECUSTOMIZE.PY

Linux/macOS: daemon polls a GitHub dead drop hourly for a second stage. Empty at release; account deleted — **anyone re-registering it could push code** to infected hosts.

Wiz observed harvested AWS role credentials tested against **Amazon Bedrock** (`ListFoundationModels`) ~4.5 hours after exfiltration — consistent with LLMjacking.

INDICATORS OF COMPROMISE

TYPE	INDICATOR
Network	<code>litellm[.]dev · www.litellm[.]dev (reg. 13 Oct) · hxxp://www.litellm[.]dev/healthcheck · path /codepy · raw.githubusercontent[.]com/letters9813/pythonclient-test/main/domain.txt · header X-Token: XWiA8gZNa7NcX2XRoAEb (fake MSIE 7.0 UA)</code>
Host	<code>site-packages/cook.pth · cook.py · unexpected sitecustomize.py · litellm-1.107.0.dist-info · /tmp/.pc_once_72129118c59f5636c0bb · /tmp/remote_code_loader.lock · /tmp/.pc_sq_* · /tmp/.pc_e_* · env PRIVCOLLECT_ACTIVE=1</code>
GitHub	<code>github[.]com/Berri-AI/litellm · commit 20535d8fallf786a6c24df7488a0e449ed5f09ad · workflow 9cf97e935aeaf791b9c8db224a2bb3ee156134ce</code>
IPs	<code>69.48.229[.]143 · 199.91.221[.]39 (LiteLLM) · 168.100.8[.]57 · 165.227.201[.]8 (Wiz)</code>
SHA-256 Linux/macOS	<code>cook.pth 7f191632e7320705d154b85204f1ae546b543acbd7e8861229b76f5042b92bdb cook.py 9dd0a3926f96a37eb86cf1ee2310d9be93139cc4ebef3ef9e2efe9531390be25 sitecustomize.py 34bef04fe88ac9d0aae0475630b1738681d15b94f233ac8aa46eacf5968305fa</code>
SHA-256 Windows	<code>cook.pth 84dc8c82c0a4a4a62552ef826b49fcc86d1a5c4a5d0f2c06665723f93d129d7d cook.py 03ac2a8b752605d1d7710979c17ff18b03dd978203ba23ba97fd8ce3d6f897ea sitecustomize.py b6122440d0d9cb1eb1ade9933f49214da8c31288de7710c076c2680dfa528465</code>
Image digests	<code>litellm index sha256:1bbcb1443bc962e45f5aad53fe3d2c30600e0379715fb6c664cdd3ac126c339e litellm-database amd64 sha256:24581ddc30767f16ecb69d2b0517556f2cd7dea56971adb42a3d1beb97d69523 · arm64 sha256:827dd893c74cf621a860a66a3a447a5c732533f087589ce905573e285c0a239c litellm-non_root amd64 sha256:645a67d810628252d63df85dcc8824c58bcc77cf8383f4d3e6321f0ccc3ce9e4 · arm64 sha256:f75c8edae59b72b1ebfd617295833f465c85bb2fab9c1d60f21b496db86cb8da</code>

Hashes match between StepSecurity and Chainguard; digests corroborated by LiteLLM. Wheel hashes for all seven platforms: see StepSecurity/Chainguard advisories.

DO NOT RUN PYTHON TO CHECK

Every Python or `pip` start re-runs the stealer on an infected host. Triage with shell tools only (page 13). `/tmp/.pc_once_*` is written only after successful exfiltration — its absence proves nothing.

RESPONSE SEQUENCE

- Kill the loader daemon; delete payload files and `__pycache__` copies.
- Rebuild — don't clean in place — pinned to `litellm==1.106.2`; pin images by digest.
- Purge caches/mirrors of 1.107.0; block `litellm.dev`; escalate any `/codepy` request.
- Rotate: instance IAM role, provider keys in the proxy DB, LiteLLM master & salt keys, `DATABASE_URL`, cloud/K8s/Vault/GitHub/npm/PyPI/SSH/AI-tool creds. Review CloudTrail for `GetCallerIdentity`.

BLUE TEAM LAB

Field commands for all four incidents

Read-only triage checks for systems you own or administer. None is exploitative. They produce clues, not verdicts — preserve evidence and escalate through your IR process.

01 · PHP INVENTORY & EXPOSURE

```
# CLI version and config (web SAPI may differ!)
php -v; php --ini
php -i | grep -E \
  'upload_progress.enabled|memory_limit|post_max'
# Which SAPIs are running?
ps -eo user,cmd | grep -E 'php-fpm|apache2|httpd|lsphp'
apachectl -M 2>/dev/null | grep -i php # mod_php = top risk
# Package level (backports keep old upstream number)
dpkg -l | grep -E '^ii +(php|libapache2-mod-php)' # Deb
rpm -qa 'php*' --last | head # RHEL
```

02 · PHP HUNTING SINCE 17 OCT

```
sudo journalctl --since 2026-10-17 -u 'php*-fpm' \
  | grep -iE 'exhausted|SIGSEGV|SIGBUS'
sudo grep -h 'Allowed memory size' \
  /var/log/apache2/error.log* 2>/dev/null | wc -l
# Bursts of HTTP 500 in access logs
awk '$9==500' /var/log/nginx/access.log | wc -l
# PHP files changed since 17 Oct (clue, not proof)
find /var/www -type f -name '*.php' \
  -newermt 2026-10-17 -ls
```

03 · EMUDHRA TRUST CHECK

```
# Is the removed root still in the system bundle?
awk -v c='openssl x509 -noout -subject' \
  '/BEGIN/{close(c)}{print | c}' \
  /etc/ssl/certs/ca-certificates.crt \
  | grep -i 'emSign ECC Root CA - G3'
# Live client test: an error is the good result
curl -sS -o /dev/null \
  https://dv-g3b-emudhra.we-have-a-problem.com/ \
  && echo TRUSTED || echo rejected
```

07 · CT WATCH FOR YOUR DOMAINS

```
# Which CAs have issued for your domain recently?
curl -s 'https://crt.sh/?q=%25.example.com&output=json' \
  | jq -r '.[].issuer_name' | sort | uniq -c \
  | sort -rn | head
# Any emSign / EM * CA issuance is suspicious now
curl -s 'https://crt.sh/?q=%25.example.com&output=json' \
  | jq -r '.[] | select(.issuer_name|test("EM |emSign")) \
  | [.not_before,.common_name] | @tsv'
```

04 · GITLAB

```
sudo cat \
  /opt/gitlab/embedded/service/gitlab-rails/VERSION
# Native crashes in the regex engine
sudo grep -iE '\[BUG\].*(onigmo|org-ruby)' \
  /var/log/gitlab/puma/puma_stderr.log
# Org-mode render timeouts
sudo grep 'Timeout::Error' \
  /var/log/gitlab/gitlab-rails/exceptions_json.log \
  | grep -c '\.org'
# Unexpected processes under the git account
ps -u git -o pid,ppid,etime,cmd \
  | grep -vE 'puma|sidekiq|gitaly|workhorse'
```

05 · LITELLM — SHELL ONLY, NEVER PYTHON

```
# Lockfiles and unpinned constraints
grep -rnE 'litellm.*(1\.\.107\.\.0|>=)' \
  --include='*.txt' --include='*.lock' \
  --include='pyproject.toml' . 2>/dev/null
# Payload files and markers
find / -xdev \( -name cook.pth -o -name cook.py \
  -o -name 'litellm-1.107.0.dist-info' \) 2>/dev/null
ls -la /tmp/.pc_once_* \
  /tmp/remote_code_loader.lock 2>/dev/null
fuser -v /tmp/remote_code_loader.lock # loader PID
# Kubernetes: running malicious digests?
kubectl get pods -A -o jsonpath='{..imageID}' \
  | tr ' ' '\n' | grep -E \
  -e '1bbcb1443bc9|24581ddc3076|827dd893c74c' \
  -e '645a67d81062|f75c8edae59b'
```

06 · EGRESS & DNS

```
# Adapt to your resolver / proxy logs
grep -E 'litellm\.dev|letters9813' \
  /var/log/squid/access.log 2>/dev/null
```

08 · VERIFY, DON'T ASSUME

CHECK	PROOF OF FIX
PHP	Advisory/package ID in changelog
eMudhra	Test host rejected by every client
GitLab	VERSION ≥ fixed release on line
LiteLLM	Rebuilt host; digest pinned; keys rotated

INTERPRET CAREFULLY

Changed files, 500s and memory errors have benign causes. Record exposure, attempt, compromise and remediation as separate facts.

LEADERSHIP / RESPONSE

What leaders actually said

Verbatim statements, with venue and date. Social-media posts are reported via multiple agreeing outlets; original posts were not independently archived.

MP

Matthew Prince

CEO, CLOUDFLARE · REUTERS INTERVIEW, OCT 2026

"...every single week it's going to feel like the sky is falling."

Prince expects a couple of turbulent years but says the industry will eventually clean much of this up. Separately, he posted that there were millions of PHP exploit attempts within hours of disclosure.

SA

Sam Altman

CEO, OPENAI · STANFORD GSB, ~23 OCT 2026

"I no longer think we're going to be able to just patch all the bugs."

Altman argued the defence model must change fundamentally, with AI itself becoming part of the shield.

JC

Jason Clinton

DEPUTY CISO, ANTHROPIC · X, AUGUST 2026

"The vulnerability tsunami is coming in just a few months..."

He warned there would not be time to patch critical infrastructure. On 22 Oct he added that things would get worse before they improved, urging security leaders to ask their boards for help now.

MB

Marc Benioff

CEO, SALESFORCE · X, 22 OCT 2026

"Jason is 100% right. Code red. [siren emoji]"

Posted in reply to Clinton — an endorsement of his warning, not a standalone policy statement. Shopify's Tobi Lütke described the same day as the wildest in cybersecurity in a long time.

THE NUMBERS BEHIND THE RHETORIC

**5,045 →
10,740**

Monthly CVEs, Jan → Aug 2026
(Google GTIG)

141

Vulns exploited Jan–Aug 2026 vs
127 in all of 2025

**50% vs
26%**

Share of AI-found flaws enabling
RCE vs overall

~66,000

FIRST forecast of 2026 CVEs

GTIG caveats: high-risk disclosures are ~3% of the total; ~5,000 Linux kernel CVEs inflate counts.

ADVISER'S VIEW

AI-enabled hacking must be treated as a speed-and-discipline problem. Organisations should reduce exposure, validate reality fast, and operationalise defence with aggressive patching, segmentation, continuous monitoring and regular incident rehearsals.

El Mostafa Ouchen, Cybersecurity Adviser — networks, digital infrastructure, education, threat intelligence.

HOW WE QUOTE

Every quotation on this page is verbatim, with venue and date, checked against the original post or recording. Paraphrase is labelled as paraphrase. Jason Clinton's title is Deputy CISO, Anthropic.

STRATEGY & ACTION PLAN

Governed AI defence, a 24-hour playbook

Replace vague calls to "use AI" with controlled, measurable workflows — and rehearse the first day of the next Black October now.

AI-ASSISTED ATTACK PRESSURE

- Faster discovery** — weeks from first bug to working RCE (PHP2Shell: 19→21 Aug).
- Pre-disclosure probing** — MadPot saw PHP2Shell traffic 4 days early.
- Scaled post-disclosure attempts** — millions within hours.
- Credential monetisation** — stolen cloud keys tested on Bedrock within ~4.5 h (LLMjacking).
- More credible lures** — AI-written phishing and lookalike repos (Berri-AI).

Context: Anthropic's Project Glasswing (Apr 2026) gave defenders early access to Claude Mythos Preview; Mozilla says it fixed 271 Firefox vulnerabilities with it (Engadget, TechRepublic). New Zealand's NCSC cautions that simply finding more vulnerabilities does not, by itself, improve security.

01 · DISCOVER

Continuous asset inventory incl. SAPI, CA bundles, SBOMs for AI gateways; authorised AI code review.

02 · PRIORITISE

Reachability × exploit evidence (KEV, honeypot data) × business criticality — not CVSS alone.

03 · VALIDATE

Human-reviewed findings; sandboxed reproduction; verify fixes by advisory/package ID, not banner.

04 · RESPOND

Pre-approved emergency change lane; rollback plans; monitored deployment.

GUARDRAILS FOR DEFENSIVE AI

Least privilege · secrets isolation · no long-lived owner tokens in agent platforms · approval gates · provenance and audit logs · rate limits · **no autonomous production changes.**

THE FIRST 24 HOURS

0-2 H

Scope

- Identify internet-facing assets running the component
- Record exposure in the ticket
- Preserve logs before rotation

2-6 H

Mitigate

- Apply vendor fix or interim config
- Confirm WAF rule in block mode
- Restrict exposure where you can't patch

6-12 H

Hunt

- Run detection rules and IOCs
- Review crashes, processes, egress
- Check secret access and new tokens

12-24 H

Close

- Verify fix by package / digest
- Rotate credentials where indicated
- Document decisions and owners

LESSONS BY INCIDENT

INCIDENT	CONTROL THAT WOULD HAVE HELPED
PHP2Shell	Runtime inventory by SAPI; retire EOL branches; proxy-level request limits as standing hygiene.
eMudhra	Track every CA bundle you ship; prefer OS trust stores with revocation; monitor CT for your domains.
GitLab	Keep DevSecOps platforms off the open internet; ship <code>puma_stderr.log</code> to the SIEM; short-lived CI secrets.
LiteLLM	Revoke on leak (not just rewrite history); fine-grained tokens; two-person release with hardware-key step-up; signed provenance; pin by hash/digest; egress allow-lists for AI gateways.

EDITORIAL INTEGRITY

Corrections log & primary sources

This edition was fact-checked against vendor advisories, root-program posts and named research through 25 October 2026. We publish our corrections openly, with dates.

FACT-CHECK LOG · CORRECTED BEFORE FIRST PUBLICATION

WAS	NOW
Wiz disclosed PHP2Shell	Calif, Wiz, OpenAI and Quarkslab co-disclosed
Cloudflare reported the 23 Oct issue	PHP project published CVE-2026-115560
eMudhra keys "compromised"	Weakly generated; computable from public certs
Removed from Mozilla trust store	OneCRL block + root removal (NSS 3.131.1)
GitLab patches 22 Oct	Released 21 Oct; CVE record 22 Oct
GitLab exploited in the wild 21 Oct	No confirmed exploitation as of 23 Oct
LiteLLM key in BerriAI/litellm .env.example	opencode.json in LiteLLM-Labs/lite-harness
PyPI/registry creds exfiltrated	Release built by LiteLLM's own pipeline
"October 23 incident"	21 Oct incident; 23 Oct write-up
docker.io/berriai/litellm	docker.io/litellm/litellm
Four leader "quotes"	Replaced with verified wording; Clinton = Deputy CISO

OPEN ITEMS — VERIFY ON THE DAY OF USE

- CISA KEV status for CVE-2026-109618/109619 and CVE-2026-110264
- Apple's eMudhra block date
- Distro package versions (from secondary aggregator)
- Whether LiteLLM has released a version newer than 1.106.2

PRIMARY SOURCES

PHP	php.net/archive/2026.php github.com/php/php-src/security/advisories/GHSA-w8hw-w9fx-j79w wiz.io/blog/php2shell-php-rce blog.cloudflare.com/php-vulnerabilities-waf-protection/ aws.amazon.com/blogs/security/amazon-threat-intelligence-identifies-php2shell-exploitation-prior-to-public-disclosure-in-madpot/ github.security.telekom.com/2026/10/php2shell-honeypot-observations.html
eMudhra	blog.mozilla.org/security/2026/10/20/protecting-firefox-users-following-the-compromise-of-emudhra-intermediate-certificates/ blog.google/security/protecting-chrome-and-android-users-from-compromised-emudhra-intermediate-ca-certificates/ bugzilla.mozilla.org/show_bug.cgi?id=2083449 emudhra.we-have-a-problem.com ctlogs.dev/blog/emudhra-ca-compromise
GitLab	docs.gitlab.com/releases/patches/patch-release-gitlab-19-5-1-released/ cert.europa.eu/publications/security-advisories/2026-019/ gitlab.com/gitlab-security-oss/tldr
LiteLLM	docs.litellm.ai/blog/security-incident-oct-21-2026-attack-chain docs.litellm.ai/blog/security-incident-oct-21-2026-stepsecurity.io/blog/litellm-1-107-0-compromised-pypi-backdoored-wheels-credential-stealer wiz.io/blog/litellm-compromised-steals-credentials socket.dev/blog/litellm-attack-timeline
Context	cloud.google.com/blog/topics/threat-intelligence/vulnerability-discovery-and-exploitation-trends-in-the-ai-era anthropic.com/glasswing ncsc.govt.nz/protect-your-organisation/frontier-ai-managing-the-increasing-risks-from-vulnerabilities/ engadget.com/ai/mozilla-says-it-patched-271-firefox-vulnerabilities...

EDITORIAL STANDARDS

Visuals are original conceptual diagrams, not leaked attack material. No exploit code is published. Quotations are verbatim with venue and date; paraphrase is marked as such. Revision history: **Rev. 1** 25 Oct 2026, first publication · **Rev. 2** 27 Oct 2026, branding update, no factual changes · **Rev. 3** 29 Oct 2026, new cover, clickable sources, defanged indicators with an attached CSV, colophon; p.15 corrected to New Zealand's NCSC and sourced the Mozilla figure.

COLOPHON

Who made this report, and how

Transparency is part of the reporting. Here is who publishes MAG212 CYBER, how this edition was produced, and how to reach us.

MASTHEAD

Publication	MAG212 CYBER — independent cybersecurity reporting, learning guides and free tools
Edition	Black October 2026 · Special Report 01 · Vol. 1 · October 2026
Publisher & editor	EI Mostafa Ouchen, Founder & Editor
Web	cyber.mag212.com
Editorial, tips & corrections	consultant@mag212.com · +1 (619) 886-9478
Published	25 October 2026 (Rev. 1)
This revision	Rev. 3 · 29 October 2026 — see the revision history on page 16
Evidence cut-off	25 October 2026. Patch status, KEV listings and versions change: verify against vendor advisories before acting.

MACHINE-READABLE INDICATORS

All indicators, hashes and SPKI fingerprints in this report are attached to this PDF as **mag212-cyber-2026-10-black-october-iocs.csv** (open your PDF reader's attachments panel). Printed indicators are defanged.

STANDARDS & DISCLOSURES

- **Evidence labels.** Confirmed: stated by the vendor, root program or named researchers with evidence. Reported: credible secondary reporting not yet confirmed. Attempted: exploitation attempts observed, success not shown. Unverified: circulating without a primary source.
- **Corrections.** We correct errors openly, with dates, in the fact-check log and revision history (page 16) and on the web edition.
- **Independence & conflicts.** MAG212 CYBER's publisher also provides network and cybersecurity consulting and seminars. Consulting work does not decide what we cover or how. No company named in this report paid for or reviewed its coverage.
- **AI use.** AI tools assisted research, drafting and layout under human editorial control. Every fact was checked against the primary sources on page 16. This edition contains no AI-generated images.
- **Safety.** No exploit code is published. Commands are read-only triage. Leak sites and stolen data are not linked.
- **Images.** Cover typography and all diagrams are original MAG212 CYBER work. Company names and marks belong to their owners and are used for identification only.

LICENCE

© 2026 MAG212 CYBER. You may share this PDF in full, unmodified, with attribution. Quotes of up to 300 words with a link to cyber.mag212.com are welcome. All other rights reserved.

FROM THE PUBLISHER · ADVERTISEMENT

Free tools and guides — email security checker, Cisco config hardening checker, subnet calculators and network cheat sheets at cyber.mag212.com/tools. **Consulting & seminars** worldwide: consultant@mag212.com.