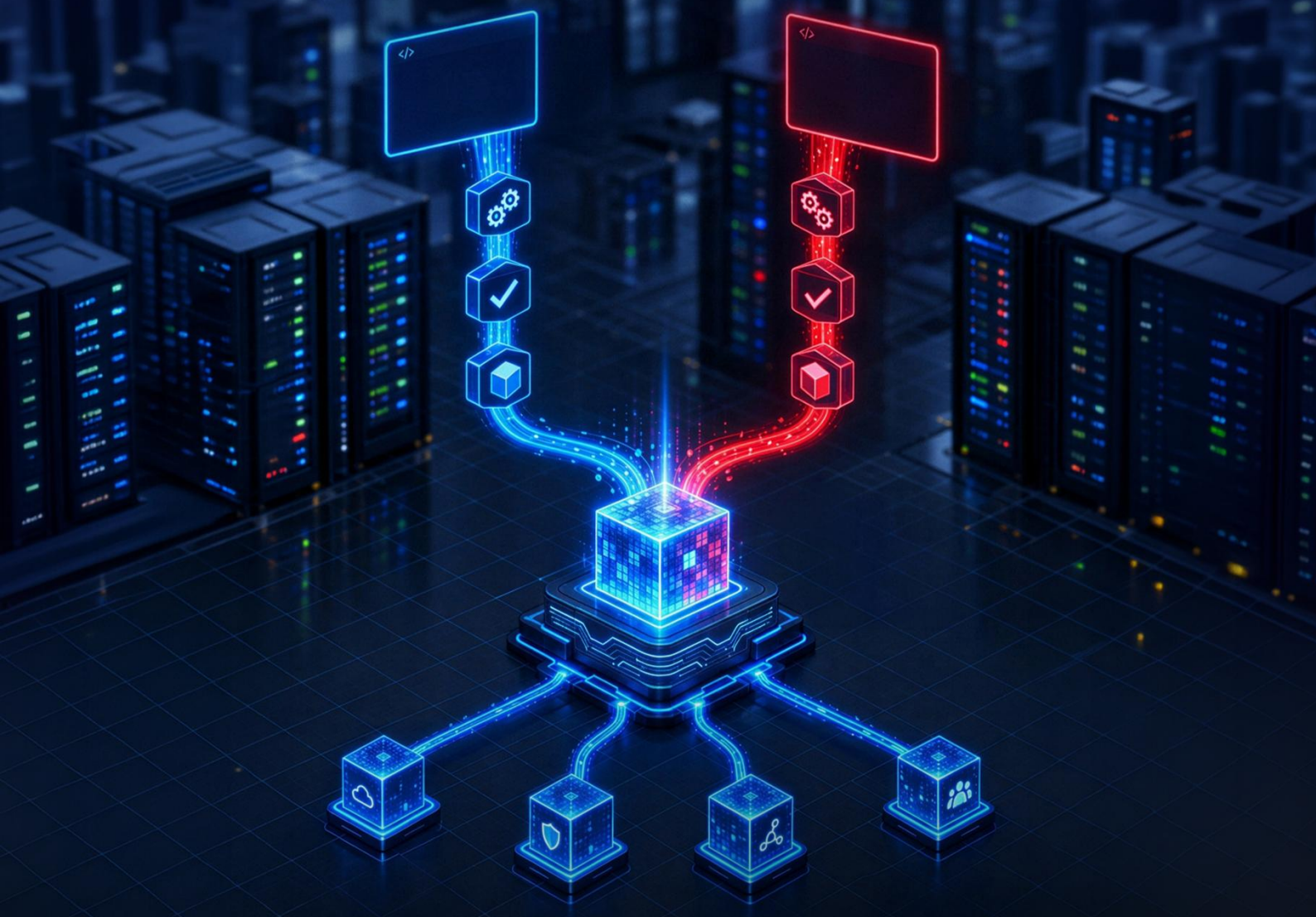
DSCI
PROMOTING DATA PROTECTION
A nasscom Initiative

DSCI THREAT INTELLIGENCE AND RESEARCH INITIATIVE

THREAT REPORT



OCTOBER 2026

Contents

Introduction....3
LiteLLM 1.107.0 supply chain compromise....4
Node.js Brotli decoder RCE (CVE-2026-111497, “Node2Shell”)....11
GitLab unauthenticated RCE via regex-engine integer overflows....17
Linux Kernel IPv4 Use-After-Free (UAF) Vulnerability....23
The Gentlemen RaaS and the Azazel Affiliate Incident....28
WhatsApp-Delivered VHDX Loader Escalates to Kernel Access....32
Threat Forecast November 2026....35

Introduction

October 2026 will be remembered as **Black October**, a month in which cyber threats converged across nation-state espionage, AI-enabled operations, software supply-chain compromises, critical infrastructure vulnerabilities and large-scale financial breaches. What made October particularly significant was not simply the volume of incidents, but the changing nature of the threat landscape. Agentic artificial intelligence moved beyond experimentation and emerged as operational offensive infrastructure that is being incorporated into real-world attack campaigns.

China-nexus threat actors such as VAPOR PANDA, leveraged the ARTEX agentic framework alongside DeepSeek to conduct coordinated operations against more than 200 organizations across multiple countries, demonstrating how AI agents could automate reconnaissance, decision-making and post-compromise activities with limited human intervention. Simultaneously, supply-chain attackers concentrated on AI developer tooling, turning trusted development ecosystems into attack surfaces, most notably through the LiteLLM compromise, while critical security incidents and vulnerabilities affected widely deployed technologies, including GitLab, Node.js and PHP through the PHP2Shell vulnerabilities. The month also witnessed major security events involving Binance, eMudhra and the reported exposure of SBI customer records, alongside the emergence of CVE-2026-98432, a critical Linux kernel vulnerability carrying a CVSS score of 9.8 under Red Hat's assessment.

Collectively, these incidents, together with the broader wave of AI-assisted attacks, infrastructure compromises and security disclosures, made October a defining moment for cybersecurity in 2026. Black October was not defined by one attack or one threat actor but the industrialization of AI-driven attacks and the exploitation of trust across the digital ecosystem.

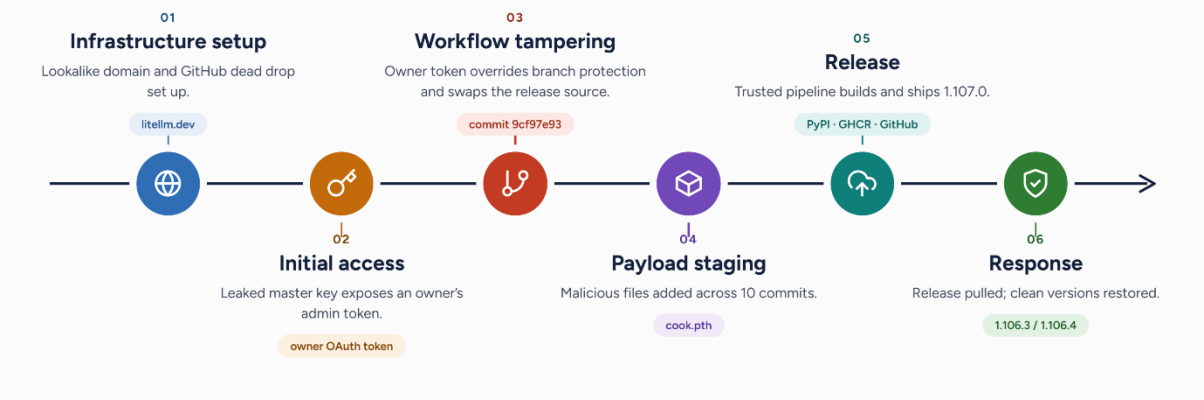
LITELLM 1.107.0 SUPPLY CHAIN COMPROMISE



Overview

On 21 October 2026 an attacker published a malicious LiteLLM 1.107.0 to PyPI and to the official Docker images. LiteLLM's own release pipeline built it from a lookalike GitHub repository (Berri-AI/litellm). Every one of the seven wheels carried two payloads, and both run whenever Python starts in the affected environment. Payload 1 is a credential stealer named privcollect, that collects browser passwords, SSH and GPG keys, cloud credentials, etc. with no import LiteLLM needed. The second payload starts a hidden background process on Linux and MacOS. Every hour it reads a GitHub file that holds address of attacker server, then downloads it and run whatever python code that server provides.

Attack Chain



Step 1: Infrastructure Setup (13 to 20 October)

The domain litellm.dev was registered on 13 October 00:45 UTC. Around six hours later, the GitHub account “letters9813” was created which hosted the dead drop. The repository containing the dead drop “pythonclient-test” was created on 18 October, added domain.txt and tested it with localhost:18909 and test.test.com:18909. On 20 October, the content of domain.txt was deleted.

Step 2: Initial Access

On 27 May 2026, opencode.json file was committed to LiteLLM-Labs repository containing a master key for LiteLLM agent platform. It was overwritten with a force push, but the key was never revoked. On October 10, the attacker used the key to call Agent Platform API which returned GitHub CLI OAuth token that was related to one of the two BerriAI’s GitHub owners. The attacker now had admin level access with that one token.

Step 3: Workflow Tampering

The branch protection rejected an unauthorized change to private release workflow, but attacker overrode it five seconds later by using owner's token and applied commit 9cf97e93, pointing the release source to Berri-AI/litellm (lookalike GitHub repository).

Step 4: Payload Staging

10 commits were added by Berri-AI <ckdi82fkoos@proton.me> in which it added:

- cook/purelib/cook.pth
- cook.py
- sitecustomize.py
- a data = "cook" line under [tool.maturin] in pyproject.toml
- COPY cook ./cook in Dockerfiles

Step 5: Release

The staging run was initiated from attacker's GitHub account (Berri-AI/litellm) on 21 October at 10:46:39. Few minutes later the production pipeline began, and the stolen token started approving the production gate which resulted in 1.107.0 wheel file built. PyPI was uploaded around 11:39:51 containing seven 1.107.0 wheels. The malicious image was published and GitHub release completed at 11:50.

Step 6: Response

Around 11:54, a third engineer's credential deleted the staging and production workflow run records. The information about the attack was reported to BerriAI in issue 47826 and the workflow was restored between 12:09 to 12:24. The malicious release 1.107.0 was removed from PyPI. Similarly, GHCR tags were removed, and GitHub removed the 'letters9813' account and malicious repository. The clean release of version 1.106.3 and 1.106.4 resumed with new image signing key.

Why didn't earlier hardening help?

After March, LiteLLM moved the release to a private repository, adopted PyPI Trusted Publishing, made release tags immutable and signed images. The attacker didn't bypass the pipeline, it used owners token which rewrote the trusted workflow, so the pipeline was built, signed and the malware was published with valid provenance. The SLSA provenance did record the lookalike source, which is how researchers traced it.

The payload



Step 1: Malicious Package Installation

The wheel metadata set *"Root-Is-Purelib"* to false, so pip downloads everything under *"litellm-1.107.0.data/purelib/*" in the root of site-packages.

Step 2: Payload Execution

On every interpreter start, Python's 'site' module runs *cook.pth* (12 bytes import cook). The file *cook.py* is a single line containing 22,251 characters that base64-decodes and zlib-decompresses a 61,000-byte stealer, privcollect, then runs it on a daemon thread. The environment variable PRIVCOLLECT_ACTIVE stops the child process from running twice. The file "*sitecustomize.py*" is loaded after the *.pth* files are processed. On Debian and Ubuntu, the Python that ships with operating system loads its own *"sitecustomize.py*" first at "*sys.path"* which can shadow the malicious one, so the attackers file may never run. This does not stop the stealer, because "*cook.pth"* runs either way.

Step 3: Sensitive Data Collection

The stealer runs 16 collection modules: *sysinfo, netinfo, browser_pw, browser_hist, ssh_keys, shell_hist, cloud_creds, dev_secrets, container_creds, cicd_secrets, ai_devtools, env_vars, db_litellm, cloud_metadata, gpg_keys* and *tls_privkey*. It targets browser password stores, SSH and GPG keys, *.pem/.key/.pfx* files, AWS/Azure/GCP/kubeconfig/Docker credentials, *.npmrc, .pypirc, .netrc, .git-credentials, .env files*, GCP/Azure metadata tokens, and config for more than 20 AI coding tools (Claude Code, cursor, Copilot, Gemini CLI, Codex, and others).

Step 4: LiteLLM Database Credential Theft

The db_litellm module starts a child python process to target LiteLLM proxy and runs

a. *SELECT "credential_name", "credential_values" FROM "LiteLLM_CredentialsTable" LIMIT 500*
b. *SELECT "model_name", "litellm_params" FROM "LiteLLM_ProxyModelTable" LIMIT 500*

After the process is complete, it kills the query engine to avoid leaving any evidence.

Step 5: Data Encryption and Exfiltration

The JSON is encrypted with AES-256-CBC, the key is wrapped with an embedded RSA-4096 key (OAEP-SHA256). A *"tar.gz"* holding *hdr.json, data.bin* and *key.bin* is posted over plain HTTP to "*http://www.litellm.dev/healthcheck"* with *X-Token: XWiA8gZNa7NcX2XRoAEb*, an X-SHA256 body hash and a spoofed MSIE 7.0 User-Agent. Only HTTP 200 counts as success, after which "*/tmp/.pc_once_72129118c59f5636c0bb*" is written. Until then, it reruns on every Python start. Captured traffic cannot be decrypted as only the holder of RSA private key can decrypt the traffic.

Step 6: Remote Code Loader (Linux and macOS)

The *"sitecustomize.py*" takes a lock on *"/tmp/remote_code_loader.lock*", forks it, calls *setsid()*, and every hour reads a hostname from *"raw.githubusercontent.com/letters9813/pythonclient-test/main/domain.txt"*. It fetches *"http://<host>/codepy*" and executes it. The domain.txt was empty at release, so the second stage did not get any instructions. Moreover, infected machines still check that GitHub file every hour. The attacker's account was deleted after detection, so if someone else registered the username *"letters9813"* and recreates that file, they could send code to every machine that is still infected. On Windows the second stage does not work at all, but the credential stealer still runs.

MITRE ATT&CK

Tactic	Technique	Procedure
Initial Access	**T1195.001:** Supply Chain Compromise: Compromise Software Dependencies and Development Tools	Malicious wheels and images published through LiteLLM's own release pipeline
	T1190: Exploit Public-Facing Application	
	T1078.004: Valid Accounts: Cloud Accounts	Stolen GitHub CLI tokens used to start and approve releases
Defence Evasion	**T1562:** Impair Defences	Owner token overrides branch protection
	T1070: Indicator Removal	Workflow run records deleted
Persistence	**T1546.018:** Triggered Execution: Python Startup Hooks	cook.pth and sitecustomize.py run at every Python start
Stealth	**T1027.013:** Encrypted/Encoded File	cook.py decodes its base64 and zlib payload at runtime
Execution	**T1059.006:** Command and Scripting Interpreter: Python	Stealer and database query run as Python code

Credential Access	**T1552.001:** Unsecured Credentials: Credentials in Files	16 modules read credential files
	T1552.004: Unsecured Credentials: Private Keys	SSH and TLS keys targeted
	T1552.005: Unsecured Credentials: Cloud Instance Metadata	Cloud metadata tokens targeted
	T1555.003: Credentials from Password Stores: Credentials from Web Browsers	Browser password stores targeted
Discovery	**T1082:** System Information Discovery	On windows, it runs systeminfo, ipconfig, route, arp and netstat
	T1016: System Network Configuration Discovery	
Collection	**T1005:** Data from local System	Gathers local files and environment data
	T1560: Archive Collected Data	Packs the data into encrypted tar.gz
Command and Control	**T1102.001:** Web Service: Dead Drop Resolver	Reads a server address from a GitHub file and fetches /codepy
	T1105: Ingress Tool Transfer	
Exfiltration	**T1041:** Exfiltration Over C2 Channel	Posts the bundle to litellm.dev/healthcheck

Indicators of Compromise (IOCs)

Type	Indicator
Package	litellm==1.107.0 (all seven wheels)
Domain	litellm[.]dev, www[.]litellm[.]dev
URL	hxxp://www[.]litellm[.]dev/healthcheck
Dead Drop	hxxps://raw[.]githubusercontent[.]com/letters9813/pythonclient-test/main/domain.txt
GitHub	github.com/Berri-AI/litellm, head commit 20535d8fa11f786a6c24df7488a0e449ed5f09ad; account letters9813
Second Stage	hxxp://<host>/codepy
Email	ckdi82fkoos@proton[.]me
Environment	PRIVCOLLECT_ACTIVE=1
Build Config	data = "cook" under [tool.maturin] in pyproject.toml
cook.pth	7f191632e7320705d154b85204f1ae546b543acbd7e8861229b76f5042b92bdb (Windows: 84dc8c82c0a4a4a62552ef826b49fcc86d1a5c4a5d0f2c06665723f93d129d7d)
cook.py	9dd0a3926f96a37eb86cf1ee2310d9be93139cc4ebef3ef9e2efe9531390be25 (Windows: 03ac2a8b752605d1d7710979c17ff18b03dd978203ba23ba97fd8ce3d6f897ea)
sitecustomize.py	34bef04fe88ac9d0aae0475630b1738681d15b94f233ac8aa46eacf5968305fa (Windows:

	b6122440d0d9cb1eb1ade9933f49214da8c31288de7710c076c2680dfa528465)
macosx_10_12_x86_64 wheel	7598727f209d717041db37f7dca271d3a7d328a279890e0e1d55375729aeb86d
macosx_11_0_arm64 wheel	0aaf7ba346d41ef2f40390436738b001180d5a1c03aa093f121078a9969d9a1a
manylinux_2_28_x86_64 wheel	462d0c9f67ca522d9e0e4acd245e90b36ce0d5353604a66bb7c43a1ec4a6fcf7
manylinux_2_28_aarch64 wheel	668169d35b87bf51324080390b1b1c5ab12bd335cfe5f1a99e3689abd5e02ae6
musllinux_1_2_x86_64 wheel	1bbce8caa94f01c0aaa0b7ddae51553a8a32e1ce53d96d44fe583272aff7ad00
musllinux_1_2_aarch64 wheel	c2e32d0ce2548aa662c18bf5001911a79fe5b30cf8357132275add02fdcde054
win_amd64 wheel	12d4ccc5c6f8d35cf4f2cf233b7f3e418e863e657d81ff5f5fc9e210005b9484
litellm-1.107.0.tar.gz (sdist, tampered config only)	2befe2a79fb4fbbd9a543b5b40b530c8cffa8771bab22b8fb22b2e4ae3b1f819

Mitigations and Recommendations

- Kill any process holding /tmp/remote_code_loader.lock (parent PID 1, working directory).

- Delete cook.pth, cook.py, sitecustomize.py and their __pycache__ entries from the shell, then rebuild the environment or image from scratch and pin litellm==1.106.2, by digest for containers.

- Purge 1.107.0 from internal mirrors, pull-through caches and uv/pip caches by the hashes above.

- From a clean machine, rotate everything the host could reach cloud keys and instance-role sessions, SSH and GPG keys, browser-saved passwords and sessions, GitHub, npm, PyPI and registry tokens, Kubernetes and Vault tokens, CI secrets and AI-tool tokens. On proxy hosts also rotate every provider key in the database, the master and salt keys, virtual keys and the DATABASE_URL password. Revoke sessions rather than assuming rotation invalidates them.

- Block litellm.dev at DNS and egress. Treat any /codepy request in logs as a wider incident.

- For maintainers: make release workflows and tags immutable, keep org-owner tokens off CI-adjacent hosts, require approvals that no single credential can satisfy, strip tokens from logged request and response bodies, and pin dependencies with hashes.

NODE.JS BROTLI DECODER RCE (CVE-2026-111497, “NODE2SHELL”)



Overview

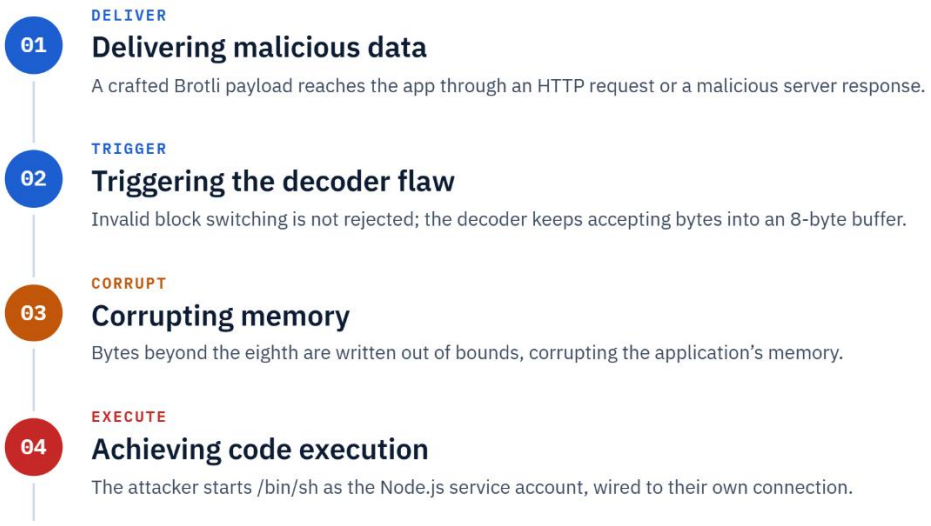
On 21 October 2026, Node.js released security updates for ten CVEs across its 22.x, 24.x and 26.x release lines. One of these, CVE-2026-111497 (Node2Shell), is a high-severity out-of-bound written in the Brotli decoder bundled with Node.js that can lead to memory disclosure, denial of service or remote code execution. Imperva's Red Team demonstrated code execution against an official Node.js 24.22.0 Linux arm64 image through both a Brotli-compressed incoming request and a consumed HTTP response. Working exploits exist for Node.js on arm64 and x86-64. One reported attempt used a public-exploit-based payload, but the final stage of that attempt reportedly failed to obtain a shell. These reports indicate active exploitation attempts, although there is no successful compromise of a victim till now. As the affected list grew to Python, Bun, .NET and others, the flaw packed up the name HTTP2SHELL.

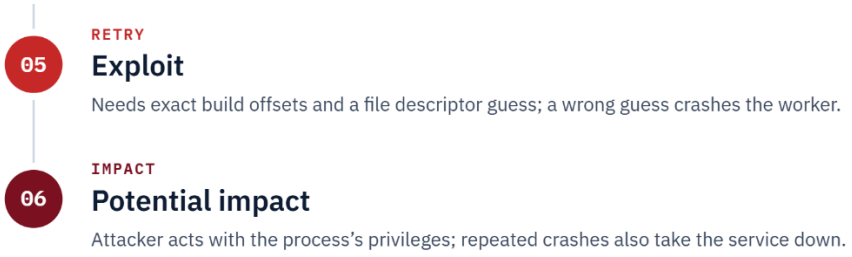
Affected and Fixed Versions

Branch	Fixed Branch
22.x	22.23.4
24x	24.22.1
26.x	26.12.1
20.x and older	No fix as 20.x reached end of life on April 30, 2026

The flaw lives in *"deps/brotli"* inside the runtime, so no npm package update fixes it and the unrelated npm package named brotli is not involved.

Attack Chain





Step 1: Delivering Malicious Data

The attacker sends a specially crafted Brotli- compressed payload to an exposed application. This can happen through an incoming HTTP request or a compressed response returned by an attacker-controlled server.

Step 2: Triggering the Decoder Flaw

When the application decompresses the payload, the Brotli decoder encounters an invalid block switching. Instead of rejecting the malformed data, the vulnerable decoder answers "I need more input" and continues processing it and keeps accepting bytes into an 8-byte internal buffer.

Step 3: Corrupting Memory

The decoder continues operating with an invalid internal state in which bytes beyond the 'eighth' are written outside the buffer causing it to write beyond the intended memory buffer. This out-of-bounds write can corrupt the application's memory.



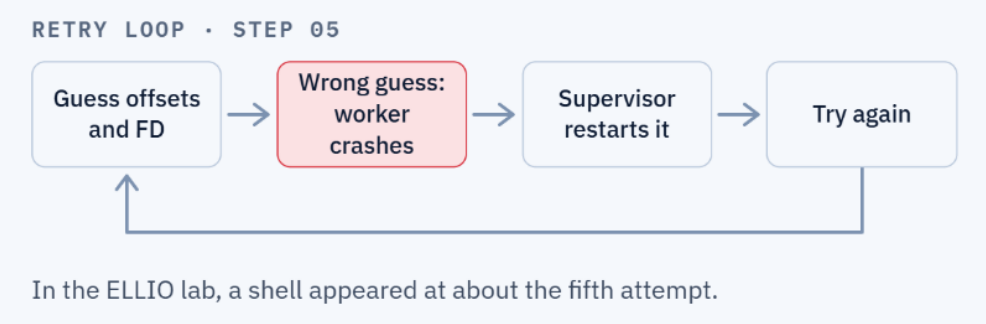
Step 4: Achieving Code Execution

Due to these suitable conditions, an attacker can exploit this and can take control of the process to start a shell (/bin/sh) running as the Node.js service account. Under certain conditions, an attacker can exploit this memory corruption to execute arbitrary code within the affected process. This gives the attacker an interactive shell on the server, controlled remotely over their own network connection.

Step 5: Exploit

The exploit needs the memory offsets of one exact Node.js build and a guess at the connection's file descriptor number. A wrong guess usually crashes the worker and a restart

supervisor then gives the attacker another try. In ELLIO's lab, a shell appeared at around the fifth attempt.



Step 6: Potential Impact

Successful exploitation could allow an attacker to act with the privileges of the compromised process, potentially exposing application data, credentials and accessible internal services. The actual impact depends on the application's permissions and environment.

Failed attempts still cause harm. Repeated crashes take the service down, and where core dumps are enabled, a crash can write live session tokens and keys into a file that others may be able to read.

Timeline

Date	Activity
21 October 2026	Node.js ships fix for 10 CVEs and google releases Brotli 1.0.10, 1.1.1 and 1.2.1
22 October 2026	Python’s Brotli and brotlicffi packages were confirmed affected
23 October 2026	Few reports show that .NET and Java were affected too
24-25 October 2026	The name HTTP2Shell start circulating and Node2Shell vulnerability was upgraded to HTTP2Shell
26 October 2026	The PHP team rebuilds its Windows builds with Brotli 1.2.1
27 October 2026	CISA adds the CVE to KEV catalog. Palo Alto confirmed PAN-OS impact

Fixed Versions

Component	Fixed In
Google Brotli	1.0.10, 1.1.1 and 1.2.1
PyPI brotli/brotlicffi	1.2.1/1.2.1.0
Bun	1.4.4 (rebuild anything made with bun build --compile)
.NET	8.0.33, 9.0.22, 10.0.14, .NET 11RC3; .NET framework is not affected
Java brotli4j	1.24.0
Envoy/Istio	Envoy 1.40.0, 1.39.3, 1.38.6, 1.37.8; Istio 1.29.9, 1.30.6, 1.31.2
Apache mod_brotli	Not affected; it compresses but does not decode

Fixes in the Same Release of Node.js Security Release

CVE	Severity	Issue	Release Line
CVE-2026-111497	High	Brotli decoder out-of-bounds write, RCE	22, 24, 26
CVE-2026-105482	High	DNS name truncation	22, 24, 26
CVE-2026-105483	High	TLS hostname verification bypass	24, 26
CVE-2026-106069	High	TLS hostname verification bypass	22, 24, 26
CVE-2026-105481	Medium	Keep-alive socket processes a response written before the request	22, 24, 26
CVE-2026-106072	Medium	Improper TLS certificate hostname validation	22, 24, 26
CVE-2026-106074	Medium	Repeated cryptographic random values in startup snapshots and single executable apps	22, 24, 26
CVE-2026-106070	Low	ZIP validation failure exposes untrusted data	26
CVE-2026-106071	Low	Permission Model filesystem access bypass	26
CVE-2026-106073	Low	fs.realpathSync permission bypass	22, 24, 26

Node2Shell was the first name, used on 21 October because Node.js shipped the first fix. The flaw actually lives in Google's Brotli decoder (versions 1.0.3 to 1.2.0), so every product that bundles that decoder inherits it.

Indicators of Compromise (IOCs)

Type	Indicator
IP	94.154.43[.]7
SHA-256	7fcbdaaecd4419e2930927844f3b772668204737a4e7602a2290c991c93c5c4b
Request header	Content-Encoding: br on POST requests
Target build	Node.js v24.22.0 (x86-64 and arm64)

Mitigations and Recommendations

- Upgrade Node.js to 22.23.4, 24.22.1 or 26.12.1 now. CISA's federal due date is 10 November. Node.js 20, 23, 25 and older get no fix, so move to a supported line.

- Update every other copy of the decoder: Brotli 1.2.1, Bun 1.4.4, PyPI Brotli 1.2.1 and brotlicffi 1.2.1.0, .NET, brotli4j 1.24.0, Envoy and Istio, and your operating system's libbrotli. Rebuild images and compiled executables, regenerate lockfiles, and restart long-running workers so the patched code loads.

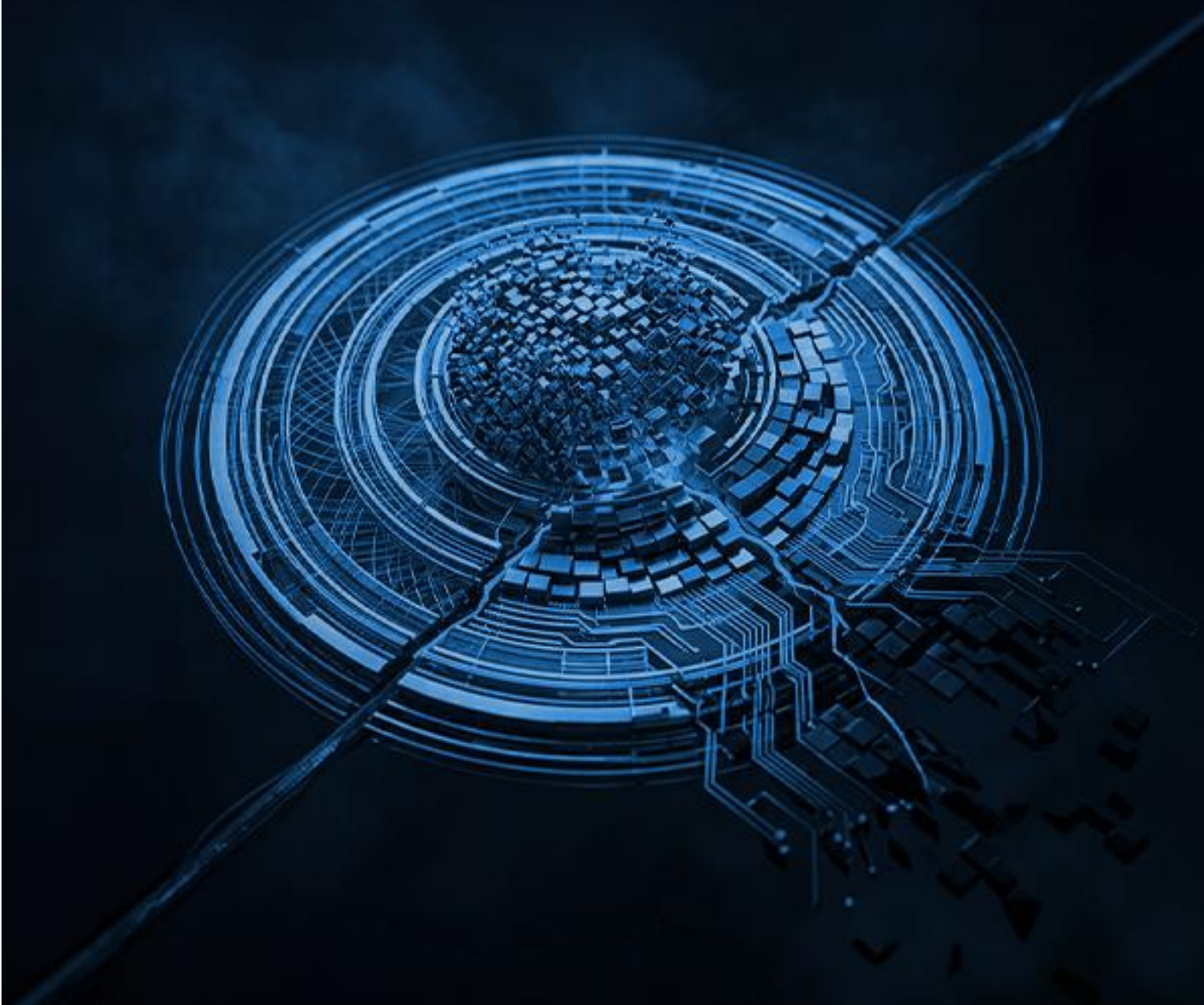
- Check the Node.js release version, not process.versions.brotli, which still reports 1.2.0 on patched builds.

- Until patched, reject Content-Encoding: br on inbound requests at the proxy or load balancer, before any parser runs. For outbound calls, remove br from Accept-Encoding and reject responses marked br before decoding, because a malicious server can send Brotli whatever the client asks. In Python, uninstalling Brotli and brotlicffi removes the vulnerable decoder if you can do without Brotli.

- Limit who can read core dumps and shorten how long they are kept. Treat repeated Brotli-related crashes as a possible attack: they hurt availability and can leak secrets.

- Hunt with the indicators above. If a Node.js service shows signs of compromise, rotate every secret that process could read.

GITLAB UNAUTHENTICATED RCE VIA REGEX-ENGINE INTEGER OVERFLOWS



Overview

GitLab shipped two maximum-severity (CVSS 10.0) fixes in ten days. CVE-2026-97229 (12 October) is an integer overflow in the CI/CD regular-expression handling; CVE-2026-110264 (21 October) is a separate integer overflow in Onigmo, Ruby's regex engine, reachable through Org-mode rendering and present since GitLab 7.0 (2014). Both were reported by researcher joaxcar through HackerOne.

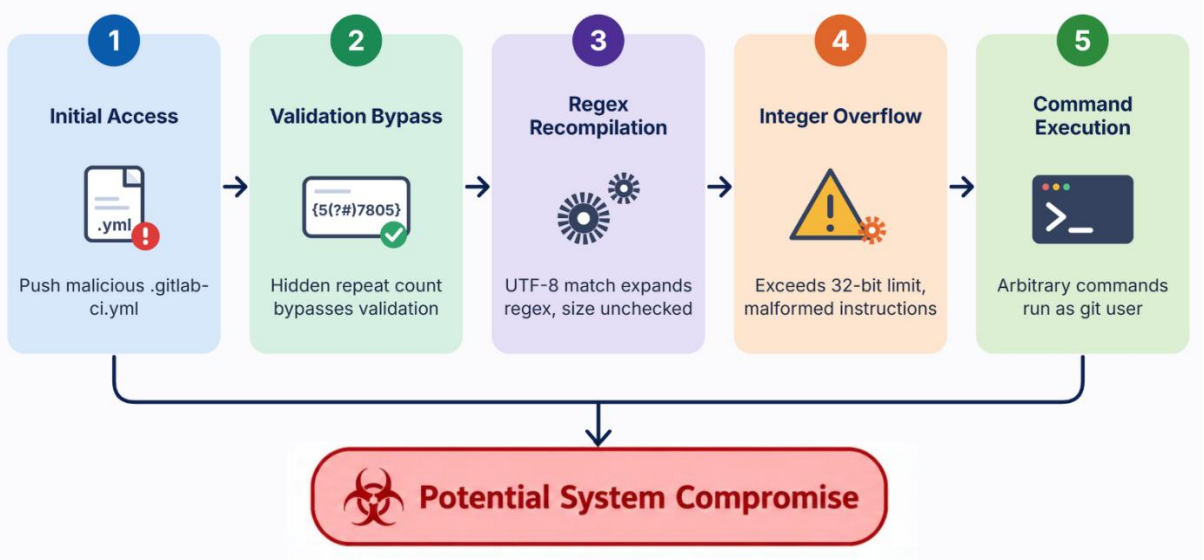
CVE-2026-97229 has been exploited against a real target. An attacker crashed the Alpine Linux project's GitLab workers eight times on 20 and 21 October, and Alpine disclosed it on 28 October. CVE-2026-110264 has public proof-of-concept code, moreover, new exploit repositories keep appearing, but with no confirmed attacks. Neither CVE was in CISA KEV as of the 28 October catalog update.

Upgrading GitLab may not fix the issue in every installation. GitLab's security fix updates the version of Ruby included in its official Linux packages and container images. However, administrators using self-compiled GitLab installations or custom images with a separately installed Ruby must update Ruby manually to a patched version (3.3.13, 3.4.12 or 4.0.9 and later). Simply upgrading GitLab will not fix the vulnerability in these setups. Ruby 3.2 is no longer supported and will not receive a security fix. Additionally, GitLab's patch announcements on 12 and 21 October did not explicitly mention this manual Ruby update requirement for self-compiled installations.

Affected Versions

Release Line	Fixed Versions
19.5	19.5.1
19.4	19.4.3
19.3	19.3.6
19.2	19.2.9
19.1, 19.0	19.1.9, 19.0.10
18.11	18.11.13
18.10 and older	No fixed release; move to a supported line

Attack Chain: CVE-2026-97229



Step 1: Initial Access

The attacker first needs the access of a public project to push a malicious *".gitlab-ci.yml"* file and a specially crafted repository path containing UTF-8 characters to a branch of that public GitHub project.

Step 2: Validation Bypass

The malicious regex hides its repeat count using an inline comment, such as {5(?#)7805}. GitLab's validation scanner reads the repeat count as 5, whereas Ruby's Onigmo regex engine ignores the comment and interprets it as 57,805, bypassing the repeat-count validation.

Step 3: Regex Recompilation

The regex is initially compiled using US-ASCII encoding and passes GitLab's validation. When it is matched against a repository path containing UTF-8 characters, Ruby recompiles it for UTF-8. During this process, six POSIX word classes *([[:word:]])* expand into large Unicode character representations, significantly increasing the compiled program size. GitLab does not recheck the size of this newly compiled regex.

Step 4: Integer Overflow

The expanded regex exceeds the signed 32-bit integer limit, triggering an integer overflow in Onigmo's program-size calculation. This results in malformed regex instructions that the attacker can manipulate to control the regex engine's execution flow and achieve operating-system command execution.

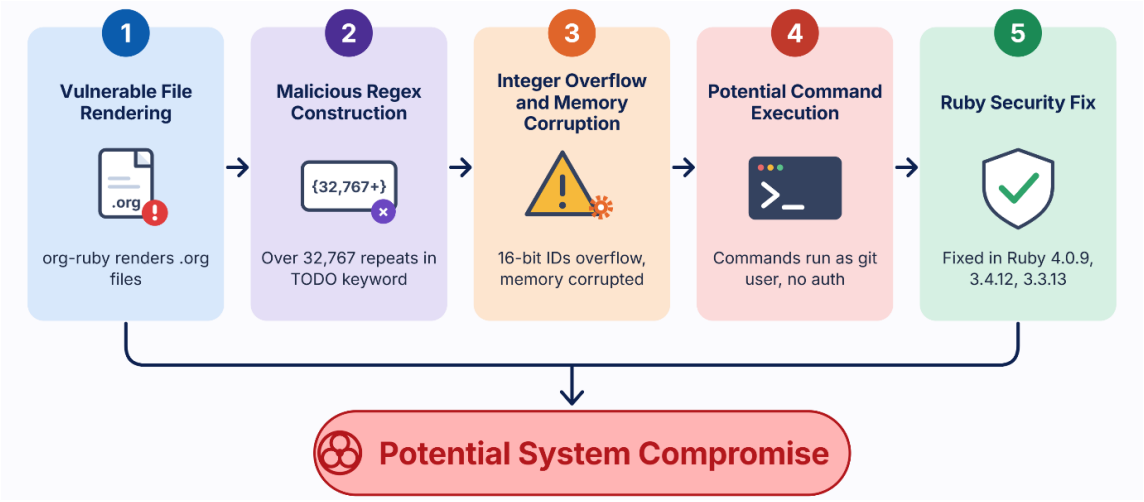
Step 5: Command Execution

Rapid7 demonstrated reliable arbitrary command execution as the GitLab git OS user using the official *"gitlab/gitlab-ce:19.4.1-ce.0"* Linux AMD64 image, even with ASLR enabled.

Exploitation in the Wild: Alpine Linux (CVE-2026-97229)

- On 28 October, the Alpine Linux disclosed that an attacker had tried to exploit CVE-2026-97229 against its own GitLab server, which hosts the repository its packages are built from.
- Alpine upgraded to GitLab 19.3.4 instead of 19.3.5 on 13 October by mistake, so the server lacked the fix.
- On 20 October, a newly registered account began probing. Over 20 and 21 October it sent crafted CI configurations that crashed GitLab workers eight times. In four of those crashes the attacker got partway through the exploit and failed at the next step. The main attack on 21 October also caused 502 errors and failed CI jobs.
- The attacker downloaded Alpine's GitLab container image. Alpine said that the attacker was trying to tune the exploit to its exact build.
- The account was blocked on 23 October. The attacker returned on 25 and 26 October and tried to exploit the vulnerability but failed.
- Alpine upgraded to 19.3.6 on 22 October, but the image used standard Ruby, so the flaw was still there. So, on 27 October, it was closed for everyone except for developers and build servers. They rebuilt it and reopened it for everyone on 29 October.

Attack Chain: CVE-2026-110264



1. **Vulnerable File Rendering:** GitLab uses the org-ruby gem to render Org-mode (.org) files into HTML. During this process, the gem constructs regular expressions using custom TODO keywords defined in the #+TODO: line without properly escaping the input.

2. **Malicious Regex Construction:** An attacker can craft a TODO keyword containing more than 32,767 regex repeat constructs. When Onigmo processes the expression, its internal 16-bit repeat IDs overflow their supported range, resulting in invalid memory references.

3. **Integer Overflow and Memory Corruption:** When GitLab matches the crafted regex while rendering the file, the overflow triggers an out-of-bounds memory write. Under certain conditions, this memory corruption can be exploited to execute arbitrary operating-system commands as the GitLab git user.

4. **Potential Command Execution:** The attack does not require authentication. A public GitLab project containing the crafted *'.org'* file can provide the necessary rendering path. GitLab's detection guidance also identifies native worker crashes and unusual Org-mode rendering errors as potential exploitation indicators. Publicly available PoCs demonstrate the underlying vulnerability and, in some cases, worker crashed. However, claims of reliable remote code execution from independent repositories should be treated cautiously unless supported by verifiable evidence.

5. **Ruby Security Fix:** Ruby fixed the underlying engine bugs on 23 October (4.0.9, 3.4.12, 3.3.13). Any Ruby application that builds a regex from attacker-controlled text ("*Regexp.new"*, interpolation) should be considered as exposed and *"Regexp.timeout"* or pattern-length limits do not provide protection against it.

Code execution as git exposes repositories, CI/CD variables, runner registration tokens, deploy keys, access tokens and every secret GitLab holds for connected cloud and registry accounts.

MITRE ATT&CK

Tactic	Technique	Procedure
Initial Access	**T1190:** Exploit Public-Facing Application	Crafted regex in a CI config (97229) or *'.org'* file (110264) overflows Onigmo; at Alpine a newly registered account sent crafted CI configs
Execution	**T1059:** Command and Scripting Interpreter	Commands run as the git user
Credential Access	**T1552:** Unsecured Credentials	Attacker reads CI/CD variables, tokens and deploy keys stored on the server
Lateral Movement	**T1550.001:** Use Alternate Authentication Material: Application Access Token	Stolen CI and registry tokens reused elsewhere (post-exploitation risk)

Impact	**T1499.004:** Endpoint DoS: Application or System Exploitation	Crafted input crashes GitLab workers (public PoCs; 8 crashes at Apline)

Mitigations and Recommendations

- Inventory every GitLab server, including Geo secondaries, staging and DR. Check the version (cat /opt/gitlab/embedded/service/gitlab-rails/VERSION, or /api/v4/version) and where its Ruby comes from.

- Upgrade within your current minor line to the fixed version above. GitLab's Linux packages and official images carry the patched Ruby. If you compile GitLab yourself or build your own image, also update Ruby to 3.3.13, 3.4.12, 4.0.9 or later; Ruby 3.2 gets no fix. Servers on 19.3.4 should move to 19.3.6. If you run 19.2 or older, plan a move to 19.3.6 or later to receive the other 11 fixes.

- Pull container images again. Any image built before it included the fixed Ruby, stays vulnerable (Alpine's alpinelinux/gitlab:19.3.6 before 27 October, and older tags).

- Until patched, remove internet exposure or restrict source ranges, and limit who can register accounts, create public projects and create personal access tokens. Restrict anonymous access to public projects or block the CI lint route. These steps narrow the unauthenticated path but do not fix the flaw.

- If compromise cannot be ruled out, treat the server as compromised: rotate the secrets in gitlab-secrets.json, all CI/CD variables, runner tokens and access tokens, and consider a rebuild as Alpine did.

LINUX KERNEL IPV4 USE-AFTER-FREE (UAF) VULNERABILITY



Overview

A critical Use-After-Free (UAF) vulnerability, tracked as CVE-2026-98432, has been identified in the IPv4 networking subsystem of the Linux kernel which is present since Linux 4.4 (2016). The vulnerability arises from a race condition during concurrent TCP connection processing, where memory containing IP socket options can be freed while another CPU is still accessing it.

An unauthenticated remote attacker could potentially exploit this condition by sending specially timed IPv4 TCP packets to a vulnerable system. Successful exploitation may result in kernel memory corruption, system crashes, Denial of Service (DoS), and potentially Remote Code Execution (RCE).

The vulnerability is particularly relevant to Linux systems accepting inbound IPv4 TCP connections, including internet-facing servers and infrastructure hosting network-accessible services. On 27 October, Anthropic's Frontier Red Team said, an unreleased internal Claude model turned the bug into a remote root exploit against servers with a Multipath TCP (MPTCP) listener.

Affected Versions

Kernel Branch	Fixed Version
Linux 7.2	7.2.10
Linux 7.3	7.3-rc7
Linux 6.12	6.12.113
Linux 6.18	6.18.56
Linux 6.6	6.6.159
Linux 6.1	6.1.190
Linux 5.15	5.15.223
Linux 5.10	5.10.272

Old kernel branches and distribution-specific releases may have separate fixes through security backports.

Red Hat has identified the following affected products:

- Red Hat Enterprise Linux 8
- Red Hat Enterprise Linux 9
- Red Hat Enterprise Linux 10
- Red Hat OpenShift Container Platform 4 and 5

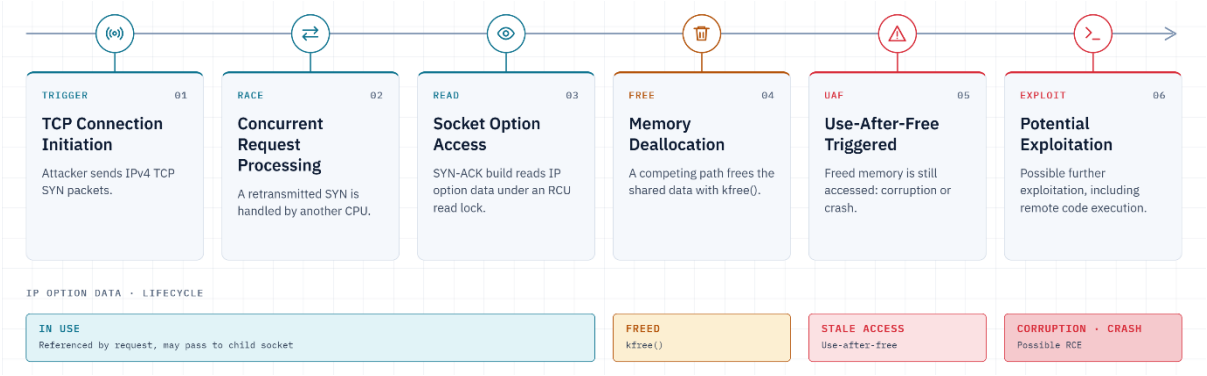
Affected kernel components:

- net/ipv4/af_inet.c
- net/ipv4/tcp_ipv4.c

Attack Chain: CVE-2026-98432

The vulnerability exists in the Linux kernel's IPv4 TCP connection-handling mechanism. During TCP connection establishment, the kernel maintains IP option data associated with a connection request socket. This data may be transferred to a child socket without copying it. The issue arises when multiple CPUs process the same connection request, allowing one execution path to access the IP option data while another frees the memory associated with it. Memory is released using kfree() operation until all readers have finished accessing it. As a result, the kernel continues accessing a memory region that is already freed, triggering a UAF condition.

Attack Flow



1. **TCP Connection Initiation:** An attacker sends IPv4 TCP SYN packets to a vulnerable Linux system to initiate a connection.

2. **Concurrent Request Processing:** While establishing the connection, the kernel processes the TCP request. A retransmitted SYN packet may be processed concurrently by another CPU, triggering another execution path for handling the connection.

3. **Socket Option Access:** While constructing the SYN-ACK response, kernel functions access IP option data associated with the connection request. This data is accessed under an RCU read lock.

4. **Memory Deallocation:** Meanwhile, another execution path transfers or releases ownership of the socket option data. If the associated request or child socket is destroyed, the kernel may free the shared memory using kfree().

5. **Use-After-Free Triggered:** The concurrent operation continues accessing the previously allocated memory even though it has already been freed. This creates a Use-After-Free condition that can result in kernel memory corruption or a system crash.

6. **Potential Exploitation:** Depending on the attacker's ability to control the affected memory and reliably trigger the race condition, the vulnerability may potentially be leveraged for further exploitation, including remote code execution.

Additional Attack Scenario – TCP Request Socket Migration:

A similar issue exists when TCP request socket migration is enabled through "*net.ipv4.tcp_migrate_req=1"* or a BPF *"BPF_SK_REUSEPORT_SELECT_OR_MIGRATE"* program. During migration, the original request socket and its cloned request socket can share the same IP option data. If one socket is destroyed while another CPU is still accessing the shared data, the kernel may free the memory, resulting in the same Use-After-Free condition.

Potential Impact

Successful exploitation of CVE-2026-98432 could have following consequences:

- **Remote Code Execution:** Potential Execution of attacker-controlled code within kernel context through successful exploitation of memory corruption.
- **Denial of Service:** Kernel crashes or system instability may make affected Linux systems unavailable.
- **Kernel Memory Corruption:** Accessing freed memory may corrupt kernel data structures and affect system stability.
- **System Compromise:** If reliable code execution is achieved, an attacker could potentially gain extensive control over the affected system.

Mitigations and Recommendations

- **Apply Kernel Security Updates:** Update affected Linux systems to kernel versions containing the security fix released by the respective Linux distributions. Prioritize internet-facing systems and critical infrastructure.

- **Implement Network-level Mitigation:** Where immediate patching is not possible, deploy firewall rules using nftables to drop incoming IPv4 TCP packets containing IP options (IP header length greater than 5). This can help reduce exposure to the vulnerable code path.

- **Review TCP Request Socket Migration:** Identify systems using *"net.ipv4.tcp_migrate_req=1"* or a BPF *"BPF_SK_REUSEPORT_SELECT_OR_MIGRATE"* program and assess their exposure to the vulnerability.

- **Identify Vulnerable Systems:** Verify their installed kernel packages and Linux servers against the security advisories published by their respective distributions. Account for security fixes delivered through backported patches.

- **Reboot Updated Systems:** Ensure affected systems are rebooted after kernel updates so that the patched kernel is loaded and actively running.

- **Validate Mitigations:** Monitor firewall activity and verify that temporary network-level restrictions do not disrupt legitimate application traffic. Remove temporary mitigation rules once the permanent kernel fix has been applied.

THE GENTLEMEN RAAS AND THE AZAZEL AFFILIATE INCIDENT



Overview

The Gentlemen is a Ransomware-as-a-Service (RaaS) group, founded by ex-Qilin affiliates in mid-2025, that offers affiliates a 90% revenue share and ranks second in 2026 by claimed victims, behind Qilin. Claims reached 683 victims by end of July 2026, and about 966 by 12 October (that last figure is unsourced). Affiliates enter through exposed edge devices or stolen credentials, steal data, disable security tools and backups, then deploy the locker, often within a day.

The Azazel incident shows the downside of that model: one affiliate used the toolkit but ran his own leak site and kept the proceeds. His exposed servers showed 24+ victim directories, about 6 TB of staged data, and the first reported use of MCP as a C2 channel.

Profile and Background

- **Origin:** The core team, ArmCorp, was a Qilin affiliate crew. It split in July 2025 after its leader ("hastalamuerte") accused Qilin's admin "Haise" of withholding about USD 48,000 from a roughly USD 200,000 negotiation. The first sample appeared on VirusTotal on 17 July 2025 with the leak-site URL already hardcoded; the RaaS opened around September 2025.
- **Leadership:** PRODAFT tracks the leader as LARVA-368 (aliases hastalamuerte, zeta88, ArmCorp), Russian-speaking. The group reportedly bars targeting of Russia and other CIS countries.
- **Model:** Double extortion, Tox-chat negotiation, and a 90% affiliate share against a 70-80% norm. A BreachForums partnership announced on 16 May 2026 recruits affiliates, pen testers and initial access brokers.
- **Malware:** Lockers for Windows, Linux, NAS, BSD and ESXi, written in Go (Garble-obfuscated on Windows) and C. Encryption is X25519 plus ChaCha20 with per-file keys. Microsoft documents a --spread worm mode and a --wipe mode.
- **Growth:** About 50 victims in late Dec 2025, 320+ in April 2026, 580 across 77 countries by 7 July, and 683 by end of July. On 23 July the group posted 30 victims in about 3.5 hours. Check Point found a SystemBC botnet of 1,570+ likely victims, far more than the leak site shows.

May 2026 Internal Leak

Around 4-5 May 2026 the administrator admitted part of the group's internal database had leaked: about 8,200 chat lines, nine internal accounts and payment histories. It showed a small, well-organized team using AI-assisted tooling, with zeta88 building and running the locker, panel and infrastructure while also operating as an affiliate. It also exposed the toolset (Mimikatz, Velociraptor, Silver, Rclone, EDR-killer collections), one negotiation that fell from USD 250,000 to USD 190,000, and stolen UK data reused against a Turkish client. The group said its core infrastructure was unaffected, and victim counts kept rising, so the operational impact looks limited.

Targets

- **Sectors:** Manufacturing leads, followed by technology and business services. Healthcare ranks third or fourth, then financial services and education.
- **Geography:** The US is the largest single target, followed by France, China, India, Vietnam and Indonesia, among 77 countries by July.
- **Notable cases:** Complexul Energetic Oltenia (Romania, energy; ERP and email down, restored from backups); Hospital Caribbean Medical Center (about 92,000 people notified); Rajagiri Hospital (India; phishing entry, 800 GB reportedly stolen); Marutake Co. (Japan, confirmed).
- **Unverified claims:** Deloitte (listed 9 Oct 2026) and KFC (listed 10 Oct 2026). Neither listing includes proof, and neither company has confirmed an incident.

Attack Chain

Stage	What happens	Tools and techniques
Initial access	Exposed edge devices, stolen credentials, phishing	FortiOS/FortiProxy CVE-2024-55591 (most cited); VPNs without MFA; access brokers; infostealer logs; CVE-2025-32433, -33073, -55182
Foothold	Covert tunnels and remote access	SystemBC, AnyDesk, Cloudflare tunnels, custom Go backdoor; scheduled tasks UpdateSystem/UpdateUser
Credential theft	Fast move to domain admin	Mimikatz, LSASS targeting, tools staged in C:\PerfLogs, PowerRun.exe
Discovery	Map network and AD	Advanced IP Scanner, SoftPerfect Network Scanner
Defence evasion	Kill security tools, clear traces	GentleKiller (8 BYOVD variants, about 400 processes), vulnerable ThrottleStop.sys (CVE-2025-7771), Defender exclusions, wevtutil
Exfiltration	Data stolen before encryption	Rclone, Restic, MinIO Client
Backup destruction	Recovery removed	200+ service-stop variants; vssadmin and wmic shadow-copy deletion
Deployment	Domain-wide push, often within a day	GPO from NETLOGON, PsExec, --spread mode; separate ESXi, Linux and NAS lockers
Extortion	Pressure to pay	Leak-site deadlines, Tox negotiation, partial leaks, data reused against victims' partners

The Azazel Incident

Azazel is a Russian-speaking Gentlemen affiliate (Russian script comments; publication server codenamed "novostnik") who breached more than two dozen organizations in six countries. He used Gentlemen's tooling, negotiation channels and ransom-note template, but posted victims on his own leak site, LEAKNED, and paid the RaaS operator nothing. His staging server held 24+ victim directories and about 6 TB, though the report does not confirm all of it was exfiltrated victim data. An unauthenticated open directory on his C2 exposed the operation.

Chain A: CI/CD secrets harvesting: Tools such as nord-stream, glato and gitleaks pulled tokens, database credentials and SSH keys from GitLab CI/CD variables and git history. Those secrets reached databases and cloud hosts, and a script (va.py) planted ransom notes on eight surfaces, including the SSH banner, the PostgreSQL cluster_name and a GitLab Issue. One SaaS breach reached 150+ databases and a dozen client companies from a single token. At a platform hosting a government-linked registry, 120,000+ records were lost and the production data directory was deleted.

Chain B: AI platform compromise: An SSRF flaw in an AI medical-imaging API gave access to the internal network. Azazel decrypted Jasypt-protected config, recovered a hardcoded JWT bypass token from git history, cracked Grafana admin hashes offline (yielding a live AWS key), and mirrored the victim's object storage to his server. A Kubernetes sweep of 6.1 TB mapped the victim's cluster. The intrusion ran for weeks and was still transferring data when discovered.

MCP abuse. CloudSEK reports this as the first confirmed live use of MCP exec_in_session as a C2 channel. A Penelope reverse-shell handler was registered as a tool in an AI coding assistant, and va.py called a local MCP endpoint (127.0.0.1:35367) to run SSH checks across six hosts. Beacon logs also show a scanner ("internet-census-mcp-scanner") hunting for exposed MCP servers.

Assessment. Azazel's tradecraft is more advanced than the core Gentlemen playbook: secrets in git history, CI/CD tokens and AI platform data are his main prizes, and one token often exposed several organizations. His careless exposure and the lost proceeds show the weak loyalty inside a 90% affiliate model.

IOCs (Azazel)

Type	IOC
C2	23.236.169[.]183
forgitlab	162.220.163[.]26
Threat actor owned GitLab hostname	forgitlab[.]com
novostnik / LEAKNED	66.179.30[.]155
Beacon check-in	141.95.252[.]30
MEGA cloud	66.203.124[.]135:443

WHATSAPP-DELIVERED VHDX LOADER ESCALATES TO KERNEL ACCESS



WhatsApp-Delivered VHDX Loader Escalates to Kernel Access

A ZIP file received over WhatsApp loaded a kernel driver on a PC. 10 seconds after it was run, it contacted its command server 13 seconds later. Treat any host that ran it as compromised.

How It Started

The infection began with a ZIP file received over WhatsApp. WhatsApp Desktop was logged in on a PC, so the file reached the machine without passing the email gateway. Inside was *"UTR_1306-10-2026.94F.vhdx"*, a virtual disk that Windows mounted as drive *"E:"*. Files in a mounted disk image carry no Mark-of-the-Web, so Windows showed no warning when *"UTR_1306-10-2026.exe"* was run.

No window or error appeared. Later, an unfamiliar program with a Chinese name was seen in Task Manager. It was signed with a certificate issued to Kafan Information Security (Shanghai), most likely stolen or misused. The PC was disconnected and the sample was reverse engineered.

What Malware Did:

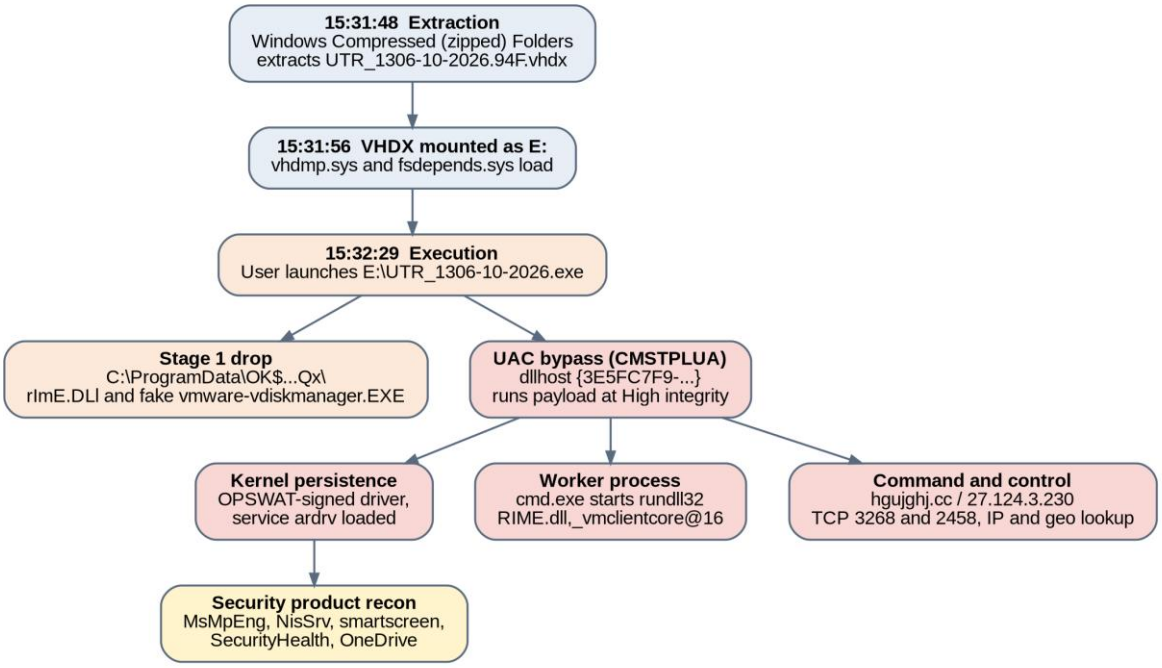
1. **Execution (15:32:29):** the program loaded an unsigned DLL, rImE.DLl.
2. **Self-copy:** it copied itself to *"C:\ProgramData\OK$%&%+,,]'#%)Qx\"* as vmware-vdiskmanager.EXE.
3. **Privilege escalation:** it used the built-in CMSTPLUA COM component to gain administrator rights without a UAC prompt.
4. **Kernel driver (15:32:38):** it loaded a signed OPSWAT driver as service `ardrv` (BYOVD), giving it kernel-level control.
5. **Reconnaissance:** it checked for Defender, SmartScreen and Security Health, and listed the webcam and microphone.
6. **Command and control (15:32:52):** it connected to *"hgujghj[.]cc"* on TCP 3268 and 2458, looked up the PC's public IP and location, and kept retrying after the host was disconnected.

A second package with the similar behavior was found later, delivered as an .iso image instead of a .vhdx. Blocking a single container format was, therefore, not enough.

Indicators of Compromise

Type	Indicator	Context
SHA-256	C74F55247AA2C72001024BD977C869CB5ADCE6F8604A4D32329F015B73AE2A49	Loader
SHA-256	5F24BB8B1B47278F3D5154627347DB255E940FE86537B4B20C4DE8E5FFDA351D	rImE.DLl implant
SHA-256	1C5129B4D63F2D5252B5B02A0A7D655DAC3B6EAC23DD4ED92B4DD55F243387FF	Driver (ardrv)
Domain	hgujghj[.]cc	C2
IP	27.124.3[.]230	C2, TCP 3268 and 2458

Type	Indicator	Context
Registry	HKLM\SOFTWARE\IpDates_i4, HKCU\Console\0	Config blobs



Recommendations

1. **Isolate and reimage:** Rebuild affected hosts from known-good media and reset every account used on them.
2. **Block and hunt:** Block the domain and IP, and search for the hashes, the ardrv service and outbound connections on TCP 3268 or 2458.
3. **Restrict messaging apps:** Limit WhatsApp desktop and similar clients on office PCs, or block file downloads through them.
4. **Block containers that can skip Mark-of-the-Web:** disk images (.iso, .img, .vhd, .vhdx) and archives some extractors open without the tag (.7z, .rar, .tar, .gz, .zip), especially when they contain executables or scripts. Windows updates since late 2022 close only to some of these gaps.
5. **Block vulnerable drivers:** Enable the Microsoft Vulnerable Driver Blocklist, with HVCI or WDAC where possible.
6. **Detect:** Alert when users mount disk images, when the CMSTPLUA DllHost.exe process starts other processes, and when a kernel service loads from a user's %TEMP% folder.

Disclaimer: This is based on analysis of one infected host (report dated 9 October 2026). This is for information only and is subject to change.

THREAT FORECAST NOVEMBER 2026



October 2026 Tactical Forecast

Forecast	Likelihood	Rationale
At least one more edge or management-plane flaw exploited within days of disclosure	Highly Likely	October's dominant pattern (FortiMail, Cisco Catalyst SD-WAN, GitLab); GitLab CVE-2026-97229 carries an FCEB remediation target around 1 November
Oracle E-Business Suite CVE-2026-101767 (CVSS 9.8) moves from honeypot probing to opportunistic mass exploitation	Likely	Only honeypot probing seen from 27 October, but public Horizon3.ai analysis exists; possible mass data-theft attempts within days-to-weeks
10 November Patch Tuesday stays well above mid-2026 volumes	Likely	October set a record near 1,400 CVEs, reportedly AI-assisted discovery; the trend is unlikely to reverse in one cycle
October's record 1,127 leak-site claims is matched or exceeded	Possible	Attacker-claimed October was up 25.8% on September; 79 active groups and operationalized AI triage sustain volume
Agentic-AI intrusions broaden across additional China-nexus clusters	Highly Likely	VAPOR PANDA hit 200-plus organizations in 11 days; CrowdStrike counted agentic-adversary attacks rising from 45 (Sept) to 126 (Oct)
Further AI developer or coding-platform breaches extend "Black October" into November	Highly Likely	Wiz documented at least nine such breaches in roughly four weeks; the Trigger.dev disclosure (1 Nov, November boundary) is an early continuation
FBI Binance attribution and 628-address release drive VASP blocklisting; Lazarus laundering shifts further into DeFi and cross-chain	Highly Likely	Formal FBI alert sits at the 31 Oct / 1 Nov cusp (November boundary) after TRM/Chainalysis attribution on 30 October
India-facing mobile banking MaaS persists, and festive-season deepfake fraud intensifies	Almost Certain	CyberDakait (17,960 devices, 7.6M SMS) and MR ROBOT are active; deepfake welfare and investment scams peak in the Nov-Dec window
More Silent Train-style retrospective disclosures of multi-year control-plane exploitation that outran remediation	Likely	Volt Typhoon's Silent Train showed years-long exploitation of control-plane devices (for example Cisco SD-WAN CVE-2026-20127, in KEV/ED-26-03 since 25 February 2026) despite listing; genuinely off-KEV exploited flaws (Roundcube CVE-2026-48842,

		Langflow CVE-2026-105740) also persist
New ransomware leak sites emerge as enforcement displaces affiliates	Likely	Law-enforcement pressure historically drives rebrands and successor brands rather than lasting volume declines

Q4 2026 Outlook (October-December)

Compressed exploitation windows. CrowdStrike's 2026 Threat Hunting Report (3 August 2026; data window July 2025 - June 2026) found 88% of proof-of-concept-available exploits used within 48 hours. The Microsoft Digital Defense Report 2026 (1 October 2026; data window July 2025 - June 2026) frames weaponization at well under 24 hours. Patch SLAs must run in hours, not days, reinforcing the KEV and forensic-triage operating model.

AI as operator, not assistant. MDDR 2026 and the GTIG/Mandiant "ARTEX and Friends" report (30 October 2026) independently converge on autonomous intrusions at machine speed. Near-term comfort: current agentic frameworks trade stealth for velocity and stay detectable through heavy forensic traffic and code-level traces, a window expected to narrow through 2027.

ERP and edge management planes stay the extortion substrate. Internet-exposed Oracle EBS and PeopleSoft app tiers join edge and VPN appliances as top mass-extortion vectors; a late-2026 or early-2027 EBS leak-site wave is plausible though none exists yet.

Holiday-season fraud starts earlier. Vendor outlooks (Infobip, reported 15 October 2026) warn AI-generated phishing, fake storefronts and synthetic identities will define the 2026 season, with overnight fraud running 1.7-2.4x midday rates. No 2026 CISA or FBI holiday-specific advisory was located at cut-off.

Date	Event	Why It Matters
3 Nov 2026	US midterm elections (November boundary)	Influence operations and election-security pressure; immediate boundary event
10 Nov 2026	November Patch Tuesday	Second Q4 patch cycle; record-volume triage likely to recur
13 Nov 2026	India DPDP Consent Manager obligations take effect (Rule 4)	First Consent Manager milestone for Indian data fiduciaries
2 Dec 2026	GitHub enforces default-off pull_request_target on public repos	Delayed from 2 November (announced 21 October); public-repo CI/CD stays an elevated surface until then

11 Dec 2026	US Cybersecurity Information Sharing Act of 2015 protections sunset unless extended	A lapse would chill private-to-government sharing that multinationals in India rely on

Analyst Assessment: Q4 risk concentrates where internet-exposed management planes meet machine-speed exploitation. It is highly likely that at least one more edge or ERP flaw is exploited within days of disclosure, and the 2 December GitHub hardening slip keeps public-repo CI/CD exposed through November. The 11 December CISA 2015 sunset is the dominant near-term policy risk.

2027 Strategic Outlook

1. **Agentic AI offense proliferates:** MDDR 2026 (1 Oct, window Jul 2025 - Jun 2026) and GTIG "ARTEX and Friends" (30 Oct) converge on machine-speed intrusion; ARTEX sustained over 1,500 commands per hour against roughly 40 for hands-on-keyboard. Spread across more China-nexus clusters through 2027 is highly likely.
2. **Vulnerability volume stays elevated and triage becomes the bottleneck:** MDDR puts H1 2026 CVEs near 40,000 and discovery-to-weaponization under 24 hours; October's record Patch Tuesday (near 1,400 CVEs) confirms the curve. Exploitation-driven rather than volume-driven triage becomes the default.
3. **AI API keys are crown-jewel assets:** "Black October" (Wiz, 30 Oct) showed a mature theft-to-resale economy, with one breach's first unauthorized model call within 30 minutes of key theft. Environment-variable hygiene and rapid key rotation become baseline controls.
4. **DPRK crypto theft is macro-scale:** TRM assessed 2026 DPRK theft at roughly USD 7.4 billion and lifetime at roughly USD 12.8 billion (30 October); the Binance heist alone was roughly USD 6.36 billion. Faster public attribution (analytics within 24-48 hours, FBI IOC drops) is likely to become the norm.
5. **Supply-chain and identity remain the control plane:** Runtime and startup-time execution displaces install-script payloads; GitHub's pull_request_target hardening (2 December) and npm's removal of bypass-2FA token direct-publish (around January 2027) push attackers toward release-workflow and maintainer-account compromise.
6. **A dense regulatory wall drives spen:** Gartner's 3Q26 forecast (24 Sep 2026) put 2026 information-security spend at USD 247.5B (+13.6%) with "Securing AI" up 68.7% into 2027. CISA 2015 sunset (11 Dec 2026), India DPDP core obligations (13 May 2027) and EU CRA main obligations (11 Dec 2027) form a converging compliance calendar.
7. **ClickFix remains the dominant initial-access TTP:** It grew about 23x from December 2025 to May 2026 with 96.3% of intrusions leading to malware, fused with

fake-AI-tool lures impersonating Claude, ChatGPT and Codex (81% of 1,258 analysed campaigns).

Post-Quantum Cryptography Track

October was the quarter where PQC moved from policy to default. NISTIR 8547 was finalized 27 October 2026 (A1), setting NIST's phase-out roadmap; OMB M-26-15 agency migration plans were due around 22 October (filing status unconfirmed); Google Cloud load balancers turned on PQC key exchange by default in October.

Date	Milestone
27 Oct 2026	NISTIR 8547 finalized; per-algorithm disallow dates set
22 Oct 2026	US agency PQC migration plans due under M-26-15 (computed)
Oct 2027	Google Cloud PQC opt-out deferral scheduled for removal
2029	Microsoft, Google, Cloudflare corporate PQC completion targets
2030 / 2031 / 2035	Federal key-establishment / digital-signature / full-migration deadlines

A forward threat signal: Lazarus Operation FlashHole (Kaspersky, disclosed 21 Oct) ran operational ML-KEM-1024 post-quantum key encapsulation in the wild since September 2024, pointing to a 2027 defensive-visibility gap where encrypted payload-fetch handshakes degrade network inspection without serving as reliable indicators.

Analyst Assessment: It is highly likely PQC requirements cascade through procurement in 2027 to suppliers, including Indian IT services and fintech firms serving US federal and G7 clients, well before any consensus Q-Day.

India & South Asia Outlook

- **Festive-season fraud surges:** Deepfake welfare and investment scams impersonating public figures on Meta platforms are almost certain to intensify through the Nov-Dec window, with likely MeitY and IT-Rules pressure on platforms over synthetic political content ahead of 2027 state elections.
- **Mobile banking MaaS scales:** CyberDakait (17,960 devices, 7.6M intercepted SMS) and MR ROBOT are highly likely to persist into 2027. Two-person approval and verified call-backs remain the only consistently effective controls against deepfake and voice fraud.
- **Cash-out-layer clampdown:** The DPDP Consent Manager obligation (13 Nov 2026) plus proposed RBI money-mule account rules (effective date unverified, cited for 1 April 2027 in vendor copy only) signal a 2027 structural squeeze on mule networks and new compliance mandates for BFSI.

- **PKI trust fallout:** As trust stores remove the emSign ECC root, legitimate Indian sites chaining to the affected eMudhra intermediates could become untrusted until reissued; expect heightened scrutiny of Indian CA key-generation practices.
- **State-nexus persistence:** Pakistan-nexus APT36/SideCopy Rust tooling with air-gap reach, and AI-accelerated China-linked credential operations, are highly likely to persist against Indian government, defense and finance targets.

Early-Warning Watch List

- Movement of Oracle EBS CVE-2026-101767 from honeypot probing to confirmed data-theft victims; a possible Clop-style or peer EBS extortion wave.
- Retrospective disclosures of multi-year control-plane exploitation despite KEV/ED listing (Volt Typhoon Silent Train; Cisco Catalyst SD-WAN CVE-2026-20127 has been in KEV/ED-26-03 since February 2026), plus genuinely off-KEV exploited flaws (Roundcube CVE-2026-48842, Langflow CVE-2026-105740).
- KEV entries for AI gateways, agent harnesses and MCP endpoints; further Black October-style AI-developer-platform breaches hunting stored API keys.
- The FBI's formal Binance attribution and 628-address IOC release; VASP blocklisting and DeFi/cross-chain laundering shifts.
- New ransomware leak sites and successor brands after KillSec and ShinyHunters-cluster enforcement.
- India: deepfake political and welfare lures, CERT-In lag against KEV flags, emSign/eMudhra reissuance notices, and leak-site posts naming Indian MSPs.

References

StepSecurity – litellm 1.107.0 Compromised on PyPI and Docker: Credential Stealer and Hourly Remote Code Loader: https://www.stepsecurity.io/blog/litellm-1-107-0-compromised-pypi-backdoored-wheels-credential-stealer

Microsoft – Inside the LiteLLM 1.107.0 supply chain compromise: Credential theft through Python startup hooks: https://www.microsoft.com/en-us/security/blog/2026/10/23/inside-the-litellm-1-107-0-supply-chain-compromise-credential-theft-through-python-startup-hooks/

Aikido – Node2Shell evolves to HTTP2Shell. Brotli flaw (CVE-2026-111497) blast radius affects many libraries & apps using HTTP: https://www.aikido.dev/blog/python-packages-vulnerable-node2shell-brotli-cve-2026-111497

Rapid7 – CVE-2026-111497: Critical out-of-bounds write in the Node.js Brotli decoder: https://www.rapid7.com/blog/post/etr-cve-2026-111497-critical-out-of-bounds-write-in-the-node-js-brotli-decoder/

Alpine Linux – GitLab security incident: attempted exploitation of CVE-2026-97229: https://alpinelinux.org/posts/gitlab-security-incident-cve-2026-97229/

Konfirmity – GitLab Critical Patch: Remote Code Execution Without Login: https://www.konfirmity.com/advisories/gitlab-critical-patch-release-19-5-1

The Hacker News – Critical GitLab Flaw (CVSS 10.0) Could Allow Unauthenticated RCE Across 12 Years of Releases: https://thehackernews.com/2026/10/critical-gitlab-flaw-cvss-100-could.html

Open Source and information security mailing list archives – Message ID:2026101531-CVE-2026-98432-d140@gregkh: https://lists.openwall.net/linux-cve-announce/2026/10/15/1

Ubuntu – CVE-2026-98432: https://ubuntu.com/security/CVE-2026-98432

RedHat Customer Portal – RHSB-2026-016 Use-after-free - kernel IPv4 (CVE-2026-98432): https://access.redhat.com/security/vulnerabilities/RHSB-2026-016

DSCI
PROMOTING DATA PROTECTION
A nasscom Initiative

ABOUT DSCI

Data Security Council of India (DSCI), set up by Nasscom, is a not-for-profit industry body and think tank on data protection, cyber security, AI Safety and critical technologies including Quantum in India. DSCI is committed to making cyberspace safe, secure, and privacy-assured, and to advancing critical technologies vital for national security through policy advocacy, thought leadership, capacity building, and outreach initiatives. DSCI engages with the government, regulators, defence, law enforcement agencies, embassies, industry members, and academia to position India as a leader in the global digital ecosystem.

Data Security Council of India
4th Floor, Nasscom Campus, Plot No. 7-10, Sector 126, Noida, UP-201303

data-security-council-of-india/ DSCI_Connect dsci.connect
dsci.connect dscivideo security chips

All Rights Reserved © DSCI 2026