n8n Under Active Exploitation: AI Workflow Platform Attack Surface

CVE-2025-68613, CVE-2026-27577, and CVE-2026-27493

Target Enterprise AI Orchestration Infrastructure

Unofficial AI-assisted Research

Cloud Security Alliance AI Safety Initiative

2026-03-12

Key Takeaways

Three critical vulnerabilities affecting the n8n workflow automation platform — CVE-2025-68613, CVE-2026-27577, and CVE-2026-27493 — are under active exploitation as of March 2026, with CISA adding the first to its Known Exploited Vulnerabilities catalog on March 11, 2026 [1]. These flaws span a critical severity range, from authenticated expression injection (CVSS 9.9) to unauthenticated remote code execution (CVSS 9.5–10.0), and affect the same application surface used by enterprises to orchestrate agentic AI workflows, large language model integrations, and automated data pipelines.

The strategic significance of compromising n8n extends far beyond the host server. n8n functions as a centralized credential vault and orchestration layer connecting OpenAI, Anthropic, AWS, Salesforce, database systems, CI/CD pipelines, and dozens of other enterprise services [2][3]. An attacker achieving code execution on an n8n instance can extract the `N8N_ENCRYPTION_KEY` environment variable and decrypt every stored credential in the platform's database, achieving lateral access to every service whose credentials are stored in the n8n instance — potentially spanning an organization's AI providers, cloud accounts, and SaaS applications — through a compact post-exploitation chain [2]. As of early February 2026, Shadowserver Foundation data indicated more than 24,700 unpatched n8n instances remained exposed on the public internet [10].

Organizations operating self-hosted n8n should treat this as an emergency patching event and initiate credential rotation for all services whose API keys or OAuth tokens are stored in n8n, regardless of whether exploitation of their specific instance has been confirmed.

Background

n8n is an open-source, self-hostable workflow automation platform analogous in function to Zapier or Make (formerly Integromat), but designed for technical users who require custom logic, code execution, and deep integrations with enterprise infrastructure. Its core value proposition is allowing non-developers and developers alike to build automated workflows — called "workflows" in n8n terminology — using a node-based visual editor supplemented by a JavaScript expression language for dynamic data manipulation [4]. n8n's architecture has made it a natural fit for organizations building AI-adjacent automation: connecting LLM calls, vector database queries, document ingestion pipelines, and API integrations that would otherwise require custom application code.

This architectural role makes n8n a high-value target. Unlike a standard web application, a compromised n8n instance provides attackers with authenticated access to every upstream and downstream system connected to its workflows. The platform stores credentials for external services in an encrypted SQLite or PostgreSQL database, with encryption managed by the `N8N_ENCRYPTION_KEY` environment variable. An attacker who achieves code execution on the n8n host can read this environment variable directly and decrypt the full credential store — a single post-exploitation step that immediately yields API keys, OAuth refresh tokens, database passwords, and cloud provider credentials for every integration configured in the platform [2].

n8n came to broad security attention in November 2025 when Cyera researchers disclosed CVE-2026-21858 (assigned 2026; disclosed November 2025), a CVSS 10.0 unauthenticated RCE vulnerability dubbed "Ni8mare" affecting approximately 100,000 self-hosted instances [3]. The Ni8mare vulnerability exploited a content-type confusion bug in n8n's webhook handling, allowing unauthenticated attackers to read arbitrary files from the host, extract admin credentials from the n8n database file, forge authentication tokens, and then execute arbitrary commands through the platform's built-in "Execute Command" workflow node [3]. A subsequent vulnerability, CVE-2026-25049, documented that the initial Ni8mare fix contained incomplete sanitization that could itself be bypassed, underscoring the systemic nature of n8n's input handling challenges. The Ni8mare disclosure set the context for the subsequent wave of critical vulnerabilities documented through early 2026, each targeting variations of the same expression evaluation and input handling subsystems.

Security Analysis

CVE-2025-68613: Expression Injection RCE

CVE-2025-68613 is an expression injection vulnerability in n8n's workflow expression evaluation system, first disclosed in December 2025 and added to CISA's KEV catalog on March 11, 2026 [1]. The vulnerability received a CVSS score of 9.9 from the CNA (GitHub), with a vector of CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H, and 8.8 from NIST's independent assessment [5]. It is classified under CWE-913 (Improper Control of Dynamically-Managed Code Resources).

The underlying issue is that n8n evaluates user-supplied expressions during workflow execution — a core feature of the platform — in a JavaScript context that is insufficiently isolated from the underlying Node.js runtime [5]. Authenticated users with workflow creation or editing permissions can craft expressions that escape the intended evaluation sandbox and execute arbitrary OS-level commands

with the privileges of the n8n process. Exploitation requires authentication but no elevated role beyond basic workflow access, a permission level that may be granted broadly in organizations that allow self-service workflow automation.

Pillar Security's subsequent research into the sandbox escape mechanism documented the specific bypass chains in detail. The expression sandbox relied on AST (Abstract Syntax Tree) rewriting to neutralize dangerous constructs, but this approach contained multiple gaps [2]. One exploit chain leveraged the `Error['prepareStackTrace']` V8 callback mechanism: by defining a custom `prepareStackTrace` function using template literal syntax to avoid detection, an attacker could invoke V8's stack frame introspection mechanism to obtain a reference to the real global `process` object outside the sandbox. Once a reference to `process` was obtained, the attacker could call `process.getBuiltinModule('child_process')` and invoke `execSync` to run arbitrary shell commands [2]. When n8n's December 23, 2025 patch blocked template literal syntax, Pillar Security documented a secondary bypass using `Object.defineProperty(Error, 'prepareStackTrace', {...})` – a function call form that the sanitizer's pattern matching did not cover [2].

Affected versions span from 0.211.0 through 1.120.3 and 1.121.0. Patched releases are 1.120.4, 1.121.1, and 1.122.0 [5]. Federal civilian agencies are required to patch by March 25, 2026 under CISA Binding Operational Directive 22-01 [1].

CVE-2026-27577 and CVE-2026-27493: March 2026 Critical Pair

On March 5, 2026, The Hacker News and SecurityWeek reported disclosure of two additional critical vulnerabilities, CVE-2026-27577 and CVE-2026-27493, representing a continued inability to fully contain JavaScript sandbox escapes in n8n's expression evaluation architecture, despite successive patch cycles [6][7].

CVE-2026-27577 carries a CVSS score of 9.4 under the CVSSv4 framework [6][12]. Researcher Eilon Cohen characterized it as "a sandbox escape in the expression compiler" in which a missing case in the AST rewriter allows the `process` object to pass through without transformation into a neutralized form [6]. Because the resulting compiled expression retains access to the Node.js `process` object, any authenticated user with workflow editing privileges can achieve full RCE on the host. The vulnerability is distinct from CVE-2025-68613 in that it exploits an omission in the AST rewriter rather than an explicitly blocked pattern, suggesting that the December 2025 patches hardened certain known attack paths without achieving comprehensive coverage of the rewriter's node handling logic.

CVE-2026-27493, scored CVSS 9.5, is considerably more severe from an attack surface perspective because it requires no authentication whatsoever. The vulnerability is a double-evaluation bug in n8n's Form nodes — a component that allows users to create publicly accessible web forms that trigger workflow execution upon submission [6][7]. Because form endpoints are intentionally public by design, they lack authentication requirements. The flaw arises from n8n evaluating form input values — including user-supplied fields like name and email — twice as expressions before sanitization occurs. This double-evaluation creates an expression injection pathway accessible to any person who can submit data to an n8n-hosted form, including anonymous internet users if the form is publicly reachable. Proof-of-concept exploitation involved submitting a standard expression injection payload in a form's name field to execute arbitrary shell commands on the n8n host [6].

When CVE-2026-27493 is combined with CVE-2026-27577, the attack chain escalates to full unauthenticated RCE on the n8n host [6]. The double-evaluation bug provides initial code execution from an unauthenticated position, and the sandbox escape provides the mechanism by which that code execution reaches the operating system. The same patch batch also addressed CVE-2026-27495 (CVSS 9.4, JavaScript Task Runner sandbox escape) and CVE-2026-27497 (CVSS 9.4, SQL injection via the Merge node), indicating the March disclosure covered a broader audit of n8n's input handling subsystems rather than isolated fixes [6][7].

All three CVEs addressed in this note affect the same version ranges: n8n versions prior to 1.123.22 in the 1.x release line, versions 2.0.0 through 2.9.2 in the 2.x line, and versions 2.10.0 through 2.10.0 in the 2.10.x line. Patched versions are 1.123.22, 2.9.3, and 2.10.1 [6].

CVE	CVSS	Auth Required	Root Cause	Patch
CVE-2025-68613	9.9	Yes (workflow edit)	Expression sandbox escape via V8 prepareStackTrace	1.120.4, 1.121.1, 1.122.0
CVE-2026-27577	9.4	Yes (workflow edit)	AST rewriter missing `process` object case	1.123.22, 2.9.3, 2.10.1
CVE-2026-27493	9.5	No	Double-evaluation of Form node input fields	1.123.22, 2.9.3, 2.10.1
CVE-2026-21858	10.0	No	Content-type confusion in webhook handler (Ni8mare)	1.121.0

CVE	CVSS	Auth Required	Root Cause	Patch
CVE-2026-25049	9.4	Yes	Incomplete sanitization bypassing prior CVE-2026-21858 fix	2.4.0

Sources: NVD [5], The Hacker News [6][10], Cyera Research [3], Pillar Security [2], SentinelOne [12]

The AI Pipeline Attack Surface

The architectural role that n8n plays in enterprise AI infrastructure amplifies the impact of any successful exploitation well beyond what a typical application compromise would entail. Modern organizations use n8n as an orchestration layer for agentic AI workflows: sequences of automated steps that invoke LLMs, retrieve documents from vector stores, call external APIs, and write results to databases or communication systems. The credentials required by these workflows — OpenAI API keys, Anthropic API tokens, AWS access keys, Azure Cognitive Services credentials, Pinecone vector database keys, and OAuth tokens for SaaS applications — are all stored in n8n's encrypted credential database [2][3].

An attacker achieving RCE on an n8n host can extract the `N8N_ENCRYPTION_KEY` environment variable and decrypt the entire credential store in a single operation. This yields not just server-level access but authenticated access to an organization's AI service spend, training data, inference outputs, and any data processed through connected workflows [2]. The Pillar Security analysis noted that attackers could additionally implement man-in-the-middle attacks against AI interactions by modifying workflows to redirect LLM API calls through attacker-controlled endpoints, allowing interception and manipulation of every prompt and response without user awareness [2]. Organizations using n8n for workflows that process regulated or sensitive data face the additional risk that prompt/response interception could expose that data without any visible disruption to workflow operation.

Cyera's Ni8mare research demonstrated the complete attack chain on a representative enterprise n8n deployment: read the SQLite database via arbitrary file read, extract the admin user's hashed password and the `N8N_ENCRYPTION_KEY`, forge a valid JWT session cookie, log in as administrator, and execute OS commands via the "Execute Command" node — all without initially holding any credentials [3]. The chain required approximately four automated steps and no specialized tooling beyond standard HTTP requests and offline cryptographic operations [3].

Supply Chain Dimension

Concurrent with the vulnerability exploitation campaign, threat actors have been observed targeting the n8n ecosystem through a supply chain vector. Endor Labs researchers documented a campaign in which eight malicious packages were published to the npm registry masquerading as legitimate n8n community node integrations [8]. Community nodes are third-party extensions that n8n users can install to add integrations not included in the core platform. These packages mimic authentic integration names — one impersonated a Google Ads integration — and exploit the fact that community nodes execute with the same process-level privileges as n8n itself.

Once installed, the malicious packages leveraged n8n's credential management system against users: they presented authentic-looking configuration screens that prompted users to enter OAuth credentials, then exfiltrated those credentials to attacker-controlled servers during workflow execution. The campaign achieved measurable adoption before removal; one package registered over 8,000 downloads [8]. An updated malicious package was published shortly before the Endor Labs disclosure [8], indicating an ongoing and adaptive campaign rather than a one-time opportunistic attempt.

This supply chain dimension underscores that the n8n attack surface is not limited to server-side vulnerabilities. Organizations permitting community node installation are exposing their n8n credential stores to third-party code that runs with privileged access to the host environment and all stored secrets.

Recommendations

Immediate Actions

The first priority for any organization operating self-hosted n8n is version verification and emergency upgrade. Organizations should confirm their running version against the patch matrix above and upgrade to 1.123.22 (1.x line) or 2.10.1 (2.x line) before any other remediation step. Upgrade completion does not constitute complete remediation: because exploitation of the sandbox escape vulnerabilities could have occurred silently — workflows continue executing normally even while modified — organizations should assume credential exposure in any instance running a vulnerable version and proceed with the second priority.

Credential rotation should be performed for every integration credential stored in n8n, including API keys for AI service providers, cloud provider access keys, database passwords, and OAuth tokens for SaaS applications. The `N8N_ENCRYPTION_KEY` itself should be regenerated, which requires re-entering all stored credentials. This step should be treated as mandatory for instances that were internet-exposed

while running vulnerable versions, absent documented compensating controls. Organizations using n8n for AI-sensitive workflows should additionally notify their AI service providers of potential key compromise to enable audit of inference logs for unauthorized usage.

Workflow logs should be audited for evidence of expression injection attempts, unexpected "Execute Command" node executions, modifications to existing workflows not attributable to known operators, and unusual outbound network connections originating from n8n. The pattern of silent workflow modification — inserting data exfiltration steps into legitimate workflows — means that log review may reveal compromise even where no outward signs of disruption occurred.

Short-Term Mitigations

Organizations should implement network-level controls to restrict internet exposure of n8n instances that do not require public accessibility. The CVE-2026-27493 Form node vulnerability requires no authentication, making any publicly accessible n8n instance with active Form-based workflows an unauthenticated RCE target regardless of patch status until the upgrade is complete. Where Form-based workflows with public endpoints are operationally necessary, organizations should place them behind a reverse proxy implementing Web Application Firewall rules blocking expression syntax patterns in form input fields.

Access control hardening within n8n should restrict workflow creation and editing privileges to the minimum set of users operationally required. The authenticated CVEs (CVE-2025-68613 and CVE-2026-27577) require workflow editing permissions; limiting this role reduces the blast radius of compromised accounts used as stepping stones. Community node installation should be disabled for all production n8n instances by setting the `N8N_COMMUNITY_PACKAGES_ENABLED` environment variable to `false`. Where community nodes are required, each package should be reviewed against npm registry metadata before installation, with particular scrutiny applied to packages with recent publication dates, unusual naming patterns, or download counts inconsistent with claimed functionality.

Runtime containment should be implemented to limit the damage achievable by any expression sandbox escape. n8n processes should run as a dedicated non-root service account with no unnecessary filesystem permissions. Where containerized deployment is in use, read-only root filesystems, capability dropping, and seccomp profiles should be applied. Outbound network access from the n8n host should be restricted to the specific endpoints required by configured integrations, blocking the exfiltration pathways used in documented attacks.

Strategic Considerations

The n8n vulnerability series reflects a structural security challenge that extends beyond any individual CVE. The recurring bypasses documented in this note suggest that AST-rewriting-based sandbox isolation is difficult to make comprehensive against a determined researcher, an observation consistent with broader security research on JavaScript sandbox design. The December 2025 patch was quickly bypassed, with Pillar Security documenting a secondary exploit chain shortly after release [2], and the March 2026 disclosure revealed that a missed case in the AST rewriter provided a fresh escape path [6]. Organizational security strategy should account for the likelihood that n8n's expression evaluation architecture will continue to yield exploitable findings as researchers examine the rewriter more deeply. This makes upgrade cadence discipline and credential-rotation preparedness more important than confidence in any specific patch.

For organizations considering long-term architectural approaches, the principle of least-privilege integration design offers more durable protection than patching velocity alone. AI workflow platforms like n8n should not be granted long-lived, high-privilege API credentials for production AI services. Where possible, credentials should be scoped to the minimum permissions required by each workflow, short-lived or rotatable on a regular schedule, and issued through secret management systems (such as HashiCorp Vault or AWS Secrets Manager) rather than stored directly in n8n's encrypted database. This approach limits the value of any individual n8n compromise to the attacker.

The public nature of n8n's threat landscape — with exploit code publicly available for CVE-2025-68613 [9] and documented Ni8mare attack chains — means that exploitation is accessible to a broad range of threat actors, not only nation-state groups. Security operations centers should add n8n-specific detection content: rules for unusual child process spawning from the n8n process, expression syntax patterns in HTTP request bodies destined for n8n form endpoints, and modifications to workflow definitions outside change management windows.

CSA Resource Alignment

The n8n vulnerability series illustrates several risk domains addressed by existing CSA frameworks and guidance.

MAESTRO Threat Model (Agentic AI): The Cloud Security Alliance's MAESTRO framework for agentic AI threat modeling identifies the orchestration layer as a high-priority attack surface [13]. n8n functions precisely as the orchestration layer in many enterprise AI deployments, and the attacker capabilities documented in this note — credential extraction, workflow modification, prompt interception

– map directly to MAESTRO's threat categories around orchestrator compromise and cross-agent privilege escalation. Organizations applying MAESTRO should assign critical risk ratings to any node in their AI system topology that functions as a multi-service credential hub.

CCM Domain: Application and Interface Security (AIS): The Cloud Controls Matrix addresses software development security and vulnerability management in its AIS domain [14]. The persistent failure of n8n's AST-rewriting sandbox to achieve lasting security under research scrutiny reflects the CCM's AIS-07 requirement for security testing throughout the development lifecycle, including adversarial review of security-critical components like expression evaluators. Cloud consumers deploying n8n as part of their AI architecture should require evidence of n8n's secure development lifecycle as part of vendor risk assessments.

CCM Domain: Supply Chain Management and Transparency (STA): The malicious community node campaign maps to CCM STA domain controls requiring software supply chain integrity verification [14]. Organizations allowing n8n community node installation should apply the same software composition analysis rigor they would apply to production dependencies, including provenance verification, behavioral analysis of new packages, and audit trails for community package installation events.

Zero Trust Architecture: The credential aggregation architecture of n8n – where a single compromised host yields credentials for dozens of external services – is antithetical to the lateral movement resistance that Zero Trust architectures are designed to enforce. CSA's Zero Trust guidance emphasizes that every service-to-service credential interaction should be independently authenticated and authorized at the point of use, with no long-lived aggregated credential stores. For AI workflow platforms specifically, this translates to preferring dynamic credential issuance over static API key storage, and ensuring that the compromise of any single orchestration platform does not yield credentials permitting unrestricted access to connected AI services.

AI Organizational Responsibilities: CSA's guidance on AI organizational responsibilities recognizes that security teams must extend their vulnerability management and incident response programs to cover the AI toolchain, not only the AI models themselves. The n8n case demonstrates that AI pipeline infrastructure presents conventional application security risks – expression injection, sandbox escape, SQL injection – that require the same remediation urgency as vulnerabilities in any other production application holding sensitive credentials.

References

1. U.S. Cybersecurity and Infrastructure Security Agency, "CISA Adds CVE-2025-68613 to Known Exploited Vulnerabilities Catalog," CISA KEV, March 11, 2026. https://www.cisa.gov/known-exploited-vulnerabilities-catalog

2. Pillar Security, "n8n Sandbox Escape: Critical Vulnerabilities in n8n Exposes Hundreds of Thousands of Enterprise AI Systems to Complete Takeover," Pillar Security Blog, February 4, 2026. https://www.pillar.security/blog/n8n-sandbox-escape-critical-vulnerabilities-in-n8n-exposes-hundreds-of-thousands-of-enterprise-ai-systems-to-complete-takeover

3. Cyera Research, "Ni8mare – Unauthenticated Remote Code Execution in n8n (CVE-2026-21858)," Cyera Research, 2026. https://www.cyera.com/research/ni8mare-unauthenticated-remote-code-execution-in-n8n-cve-2026-21858

4. n8n, "n8n Documentation," n8n.io, 2026. https://docs.n8n.io

5. National Vulnerability Database, "NVD – CVE-2025-68613," NIST, 2025. https://nvd.nist.gov/vuln/detail/CVE-2025-68613

6. Ravie Lakshmanan, "Critical n8n Flaws Allow Remote Code Execution and Exposure of Stored Credentials," The Hacker News, March 5, 2026. https://thehackernews.com/2026/03/critical-n8n-flaws-allow-remote-code.html

7. SecurityWeek, "Critical N8n Vulnerabilities Allowed Server Takeover," SecurityWeek, March 2026. https://www.securityweek.com/critical-n8n-vulnerabilities-allowed-server-takeover/ (URL returned 403 as of March 12, 2026; claims corroborated by [6] and [12])

8. Ravie Lakshmanan, "n8n Supply Chain Attack Abuses Community Nodes to Steal OAuth Tokens," The Hacker News, January 2026. https://thehackernews.com/2026/01/n8n-supply-chain-attack-abuses.html

9. Censys, "December 22 Advisory: Critical n8n Vulnerability Allows Remote Code Execution [CVE-2025-68613]," Censys, December 22, 2025. https://censys.com/advisory/cve-2025-68613/

10. Shadowserver Foundation / The Hacker News, "CISA Flags Actively Exploited n8n RCE Bug as 24,700 Instances Remain Exposed," The Hacker News, March 2026. https://thehackernews.com/2026/03/cisa-flags-actively-exploited-n8n-rce.html

11. SOCRadar, "CVE-2025-68613: Critical RCE Vulnerability Disclosed in n8n Workflow Automation," SOCRadar, 2025. https://socradar.io/blog/cve-2025-68613-rce-n8n-workflow-automation/ (URL returned 403 as of March 12, 2026)

12. SentinelOne, "CVE-2026-27577: n8n Workflow Automation RCE Vulnerability," SentinelOne Vulnerability Database, 2026. https://www.sentinelone.com/vulnerability-database/cve-2026-27577/

13. Cloud Security Alliance, "MAESTRO: Threat Modeling AI Agents and Agentic AI Systems," CSA AI Safety Initiative, 2025. https://cloudsecurityalliance.org/research/working-groups/artificial-intelligence/

14. Cloud Security Alliance, "Cloud Controls Matrix v4.0," CSA, 2021 (updated 2024). https://cloudsecurityalliance.org/research/cloud-controls-matrix/

