CSAI Foundation

Cloud Security Alliance AI Safety Initiative

n8n Weaponized for Phishing and Device Fingerprinting

How Threat Actors Abuse AI Workflow Automation Infrastructure at Scale

Unofficial AI-assisted Research

2026-04-16

Key Takeaways

- Cisco Talos documented a sustained campaign abusing n8n webhook infrastructure from October 2025 through March 2026, with March 2026 email volume approximately 686% higher than January 2025 baseline levels [1][2].
- Threat actors exploit n8n webhooks for two distinct purposes: delivering malware payloads behind CAPTCHA gates while impersonating trusted cloud services, and fingerprinting email recipients through invisible tracking pixels to map target infrastructure [1].
- Two critical vulnerabilities — CVE-2026-21858 (CVSS 10.0, unauthenticated RCE) and CVE-2025-68613 (CVSS 9.9 per the GitHub CNA, authenticated expression injection) — create an attack surface independent of the social engineering abuse; CISA added CVE-2025-68613 to its Known Exploited Vulnerabilities catalog in March 2026 [3][4][5].
- A concurrent supply chain attack distributed at least nine malicious npm packages mimicking n8n community nodes, harvesting OAuth tokens and API credentials from workflow operators [8].
- Based on these findings, CSA recommends that organizations running self-hosted n8n instances treat them as high-value attack surfaces requiring the same rigor applied to public-facing application infrastructure.

Background

n8n is an open-source, "fair-code" licensed workflow automation platform designed to connect hundreds of services, APIs, and AI models through a visual node-based interface [9]. Originally conceived as a self-hosted alternative to commercial automation services such as Zapier and Make, n8n gained substantial adoption among technically sophisticated teams for its flexibility: users can write custom JavaScript or Python transformations, integrate with over 400 services, and build AI-native pipelines through native LangChain support and connections to large language models [9]. The platform can be deployed on-premises, in private cloud environments, or accessed via n8n's cloud offering, with the self-hosted option particularly prevalent in enterprise and developer communities where data sovereignty is a concern.

The architecture that makes n8n powerful is the same architecture threat actors have learned to exploit. At its core, n8n exposes externally accessible webhook endpoints that listen for inbound HTTP requests and trigger downstream workflow logic. These webhooks are a fundamental mechanism for integrating with third-party services that push events — payment processors, version control systems, SaaS platforms — and their URL structure does not visually distinguish n8n's own infrastructure from content delivered on behalf of an attacker. By mid-2025, n8n had accrued a substantial installed base of publicly accessible deployments, a growth trajectory reflected in later scanning data showing tens of thousands of internet-facing instances by early 2026 [4]. This combination of legitimate trust, broad connectivity, and externally accessible endpoints made the platform an attractive proxy layer for large-scale phishing operations.

Security Analysis

Webhook Abuse for Phishing and Malware Delivery

Beginning in October 2025 and intensifying markedly through early 2026, Cisco Talos documented a campaign in which threat actors embedded n8n-hosted webhook URLs in phishing emails impersonating shared Microsoft OneDrive file notifications [1]. The social engineering premise was notably mundane — likely a deliberate choice to avoid suspicion — with emails appearing to alert recipients that a colleague had shared a document and that clicking would open it. The actual destination was an n8n webhook URL under attacker control.

The webhook mechanism provided threat actors with two critical operational advantages. First, webhooks mask the origin of the content they deliver — from the recipient's perspective, the link resolves to a legitimate-appearing domain rather than an attacker-controlled server, lending the lure a degree of implicit credibility. Second, n8n webhooks can be programmed to inspect incoming HTTP request headers and dynamically serve different responses based on user-agent strings, geographic signals, or other session attributes. This capability allows phishing operators to selectively present malicious content only to targeted profiles while returning benign responses to automated scanning infrastructure, significantly complicating detection [1].

Recipients who clicked the embedded links were directed to a CAPTCHA page hosted through the n8n workflow chain. Upon completing the CAPTCHA — a step intended to verify human interaction and frustrate automated analysis — a download button appeared triggering retrieval of a malicious executable from an attacker-controlled external host. In the primary observed variant, this payload was a file named `DownloadedOneDriveDocument.exe`, structured as a self-extracting archive [1].

Once executed, it installed a modified version of the Datto Remote Monitoring and Management (RMM) tool, configured it as a persistent scheduled task, and executed a chain of PowerShell commands that established a command-and-control connection through Datto's legitimate `centrastage[.]net` relay domain [1]. By routing control traffic through a recognized vendor's infrastructure, the malware is designed to complicate network-layer detection.

A secondary variant observed in the same period delivered a maliciously modified Windows Installer file named `OneDrive_Document_Reader_pHFNwtka_installer.msi`, protected by the Armadillo anti-analysis packer. This installer deployed the ITarian Endpoint Management RMM tool in a similarly backdoored configuration, with accompanying Python modules performing credential and data exfiltration [1]. The use of multiple legitimate RMM platforms as post-compromise control mechanisms is consistent with a pattern observed across multiple unrelated campaigns in which attacker-controlled access blends into the noise of routine IT management traffic.

Device Fingerprinting via Tracking Pixels

In parallel with malware delivery campaigns, Talos observed a distinct use of n8n webhooks for passive reconnaissance: embedding invisible tracking pixels in email bodies [1]. When an email client renders an HTML message, it automatically issues an HTTP GET request to any referenced image URL in order to load the image. By substituting an n8n webhook URL as the image source and appending tracking parameters — such as the recipient's email address or an internal target identifier — in the URL, attackers receive a notification at the moment the email is opened. This notification confirms the address is active, identifies the email client and operating system through the user-agent string, and in some configurations records the originating IP address, enabling rough geographic or organizational attribution.

Device fingerprinting of this type imposes no interaction requirement on the target. Unlike credential-harvesting pages that require a click, tracking pixel collection occurs automatically upon email render. For threat actors conducting large-scale reconnaissance prior to more targeted operations, this approach would efficiently separate active, monitored inboxes from dormant or invalid addresses and provide basic signals about the technical environment of each target.

Critical Vulnerabilities in the n8n Platform Itself

Independent of the social engineering abuse, security researchers disclosed multiple critical vulnerabilities in n8n between late 2025 and early 2026 that provide a direct attack path to organizations running the platform.

CVE-2025-68613 is an expression injection vulnerability in n8n's expression evaluation engine, carrying a CVSSv3.1 score of 9.9 (per the GitHub CNA) [5]. n8n workflows support dynamic expressions — inline code evaluated at runtime to extract and transform data — and the affected versions failed to properly sandbox this evaluation, allowing an authenticated user to inject malicious code that executes in the context of the n8n server process. Active exploitation was confirmed in mid-January 2026, when Intel 471 reported that Akamai's security intelligence and response team had first observed the Zerobot botnet leveraging the flaw to compromise exposed instances [6]. CISA added CVE-2025-68613 to its Known Exploited Vulnerabilities catalog on March 11, 2026, setting a remediation deadline of March 25, 2026, for federal agencies [4]. As of mid-March 2026, more than 24,700 n8n instances remained unpatched and exposed online, according to Shadowserver Foundation scanning data [4].

CVE-2026-21858, dubbed "Ni8mare" by the discovering researchers at Cyera, carries a CVSS score of 10.0 — the maximum possible [3]. Unlike CVE-2025-68613, this vulnerability requires no authentication: it exploits improper handling of webhook requests to achieve unauthenticated remote code execution, granting an attacker full control over the n8n instance without any credentials. The two vulnerabilities can be chained — CVE-2026-21858 provides unauthenticated initial access, while CVE-2025-68613 enables deeper code execution in specific version ranges — further lowering the barrier to full compromise [7]. A public Metasploit module for CVE-2025-68613 was also made available, placing reliable exploitation within reach of threat actors with limited technical sophistication [6].

Rapid7's post-exploitation research identified a broader cluster of vulnerabilities disclosed in a single batch, collectively exposing n8n to remote code execution, command injection, and arbitrary file access through authenticated functionality [7]. Across these flaws, a consistent underlying pattern emerged: insufficient isolation between workflow execution contexts, configuration storage, and the underlying host operating system.

The following table summarizes the principal CVEs affecting n8n at the time of publication:

CVE	CVSS	Auth Required	Impact	Status
CVE-2026-21858 ("Ni8mare")	10.0	No	Unauthenticated RCE	Patched
CVE-2025-68613	9.9	Yes	Expression injection / RCE	CISA KEV (added Mar 11, 2026)
N8scape (cluster)	High	Yes	RCE, command injection, file access	Patched in recent versions

Supply Chain Attack on Community Nodes

A concurrent threat vector targeted the n8n ecosystem at the package level rather than the runtime. n8n's extensibility relies on "community nodes" — custom integration packages published to the npm registry that users can install directly into their n8n instances. In early 2026, security researchers at Endor Labs identified a campaign in which threat actors published at least nine malicious npm packages masquerading as popular n8n community node integrations, including packages impersonating the Google Ads integration [8]. The packages were attributed to multiple pseudonymous accounts and designed to trigger credential harvesting during legitimate workflow execution rather than at installation time.

The fundamental security risk is architectural: community nodes run with the same process-level access as n8n itself [8]. As of early 2026, there is no sandboxing or execution isolation between user-installed node code and the n8n runtime. A malicious community node can read environment variables (including stored credentials), access the local file system, make arbitrary outbound network requests, and receive decrypted OAuth tokens and API keys as they flow through workflow execution. When a victim configured one of the fraudulent integrations and ran a workflow, credentials were silently exfiltrated to an attacker-controlled server without generating errors or visible anomalies in workflow output.

Multiple malicious packages have been removed from the npm registry; one cluster is tracked under GHSA-77g5-qpc3-x24r [11]. However, organizations that installed affected packages before removal may retain compromised credentials in downstream services.

Recommendations

Immediate Actions

Security teams should treat n8n instances as high-risk perimeter systems and act on the following priorities without delay.

Organizations running self-hosted n8n instances must immediately verify patch status against CVE-2026-21858 and CVE-2025-68613. Both vulnerabilities have confirmed in-the-wild exploitation and available public exploit tooling. Instances that cannot be patched immediately should be isolated from public internet access pending remediation. The n8n security advisory from January 2026 documents the affected version range (1.65–1.120.4, fixed in version 1.121.0) [10]; the NVD entries for CVE-2025-68613 [5] and CVE-2026-21858 [3] provide additional CVE-specific remediation detail.

Email security controls should be updated to flag or quarantine messages containing webhook URLs from automation platforms including n8n. The `webhook.n8n.cloud` domain and self-hosted subdomain patterns associated with n8n webhook endpoints should be evaluated in email gateway rules and URL reputation systems. Security teams should review email logs for historical traffic to n8n webhook domains to assess prior exposure.

Credential rotation is warranted for any organization whose n8n instance may have been exposed to CVE-2025-68613 or CVE-2026-21858, as well as for any organization that installed community node packages from the affected npm accounts. OAuth tokens, API keys, and service account credentials stored in n8n's credential vault should be treated as potentially compromised.

Short-Term Mitigations

After addressing the immediate patching and credential priorities, organizations should audit their n8n deployments against a broader set of security posture controls.

Network access to n8n instances should be restricted to the minimum necessary audience. Public-facing deployments — particularly those with anonymous webhook access enabled — present the largest attack surface. Where workflows do not require external webhook ingestion, disabling or IP-restricting the webhook interface significantly reduces exposure. Authentication should be enforced on all n8n deployments, including internal ones, as the CVE-2025-68613 authentication requirement does not represent a meaningful barrier when credential compromise is factored in.

Community node installation should be gated by a formal review process. Given the absence of runtime sandboxing, any community node carries the same trust level as the n8n platform itself. Organizations should maintain an approved node allowlist and audit all currently installed community nodes against it, verifying publisher identity and npm registry provenance. Automated dependency scanning tools should be integrated into the node management workflow.

Device fingerprinting via tracking pixel is difficult to detect at the mail delivery layer because the pixel URL may appear syntactically normal. End-user awareness training should advise against enabling automatic image loading in email clients that render HTML by default, and mail client configuration baselines should be reviewed to ensure auto-load is not universally enabled across the organization.

Strategic Considerations

The n8n abuse is consistent with an emerging pattern in which legitimate, developer-oriented infrastructure becomes an attractive proxy layer for threat actors. Automation platforms, webhook services, and low-code integration tools occupy a trust gap: they are legitimate services whose URLs and

domains are not inherently suspicious, yet they can be configured by any registered user to serve attacker-controlled content. Organizations adopting workflow automation platforms — including n8n, Zapier, Make, and similar tools — should evaluate security controls not only on the platform itself but on the content and destinations those platforms can broker.

From a threat modeling perspective, self-hosted automation platforms that support inbound webhook triggers and outbound internet connectivity should be treated analogously to edge proxies or API gateways: as components capable of bridging internal resources to the public internet with attacker-configurable routing. This framing argues for network segmentation that prevents n8n instances from reaching sensitive internal systems unless those paths are explicitly authorized and monitored.

The supply chain dimension underscores the need for software bill of materials (SBOM) practices to extend to operational platforms, not just application code. When a workflow automation platform supports extensibility through third-party packages, those packages should be subject to the same vetting applied to application dependencies.

CSA Resource Alignment

The threat patterns documented in this note map directly to several layers of CSA's published guidance for AI and cloud security.

MAESTRO (Multi-Agent Threat Modeling): n8n's agentic AI workflow capabilities — multi-step automation chains connecting LLMs, APIs, and data stores — fall within MAESTRO's scope for agentic AI threat modeling [12]. Layer 6 (Operations and Processes) of the MAESTRO framework addresses the operational risks introduced when automated agents execute code and make outbound connections with limited human oversight. The n8n webhook abuse and expression injection vulnerabilities illustrate the risk class MAESTRO identifies as "unrestricted action execution," in which a workflow agent's connectivity exceeds the scope of its intended function.

CSA AI Controls Matrix (AICM): The AICM's controls for AI Supply Chain (AI-SC) and Identity and Access Management (AI-IAM) apply directly to the community node supply chain attack [13]. The AICM's guidance on dependency provenance and runtime isolation maps to the architectural gap that allowed malicious community nodes to access the full credential scope of the n8n runtime. Additionally, AICM's AI Incident Response (AI-IR) controls should be referenced when scoping the investigation following a potential n8n compromise.

CSA Zero Trust Guidance: The use of n8n as a proxy layer highlights the limits of perimeter-based trust [16]. Zero Trust principles — verify explicitly, use least privilege access, assume breach — apply to internal automation platforms as much as to external-facing services. n8n instances should not be implicitly trusted to bridge internal network segments, and all connections from automation platforms to internal systems should be explicitly authorized, authenticated, and logged.

CSA Cloud Controls Matrix (CCM): CCM domains relevant to this threat include Supply Chain Management and Transparency (STA), Infrastructure and Virtualization Security (IVS), and Identity and Access Management (IAM) [14]. Organizations using CCM as their primary compliance framework should evaluate n8n deployments against these control families when performing cloud security assessments.

STAR (Security, Trust, Assurance and Risk): Organizations that have registered self-hosted n8n deployments or third-party integrations in the STAR registry should review their published security disclosures in light of these vulnerabilities [15], particularly if those deployments process customer or regulated data.

References

[1] Cisco Talos Intelligence. "The n8n n8mare: How threat actors are misusing AI workflow automation." Talos Intelligence Blog, April 2026.

[2] Ravie Lakshmanan. "n8n Webhooks Abused Since October 2025 to Deliver Malware via Phishing Emails." The Hacker News, April 2026.

[3] Cyera Research. "Ni8mare – Unauthenticated Remote Code Execution in n8n (CVE-2026-21858)." Cyera Research Blog, 2026.

[4] Ravie Lakshmanan. "CISA Flags Actively Exploited n8n RCE Bug as 24,700 Instances Remain Exposed." The Hacker News, March 2026.

[5] NIST National Vulnerability Database. "CVE-2025-68613 Detail." NVD, 2026.

[6] Intel 471. "CVE-2025-68613: Zerobot Botnet Exploits Critical Vulnerability Impacting n8n AI Orchestration Platform." Intel 471 Blog, 2026.

[7] Rapid7 Attack Research. "Ni8mare and N8scape Flaws Among Multiple Critical Vulnerabilities Affecting n8n." Rapid7 Blog, 2026.

[8] Endor Labs. "N8mare on Auth Street: Supply Chain Attack Targets n8n Ecosystem." Endor Labs Blog, 2026.

[9] n8n. "Workflows App Automation Features." n8n.io, 2026.

[10] n8n Blog. "Security Advisory: Security Vulnerability in n8n Versions 1.65–1.120.4." n8n Blog, January 2026.

[11] GitHub Advisory Database. "GHSA-77g5-qpc3-x24r." GitHub, 2026.

[12] Cloud Security Alliance. "Agentic AI Threat Modeling Framework: MAESTRO." CSA, February 2025.

[13] Cloud Security Alliance. "AI Controls Matrix." CSA, 2025.

[14] Cloud Security Alliance. "Cloud Controls Matrix." CSA, 2024.

[15] Cloud Security Alliance. "STAR: Security, Trust, Assurance and Risk." CSA, 2026.

[16] Cloud Security Alliance. "Zero Trust Working Group." CSA, 2026.