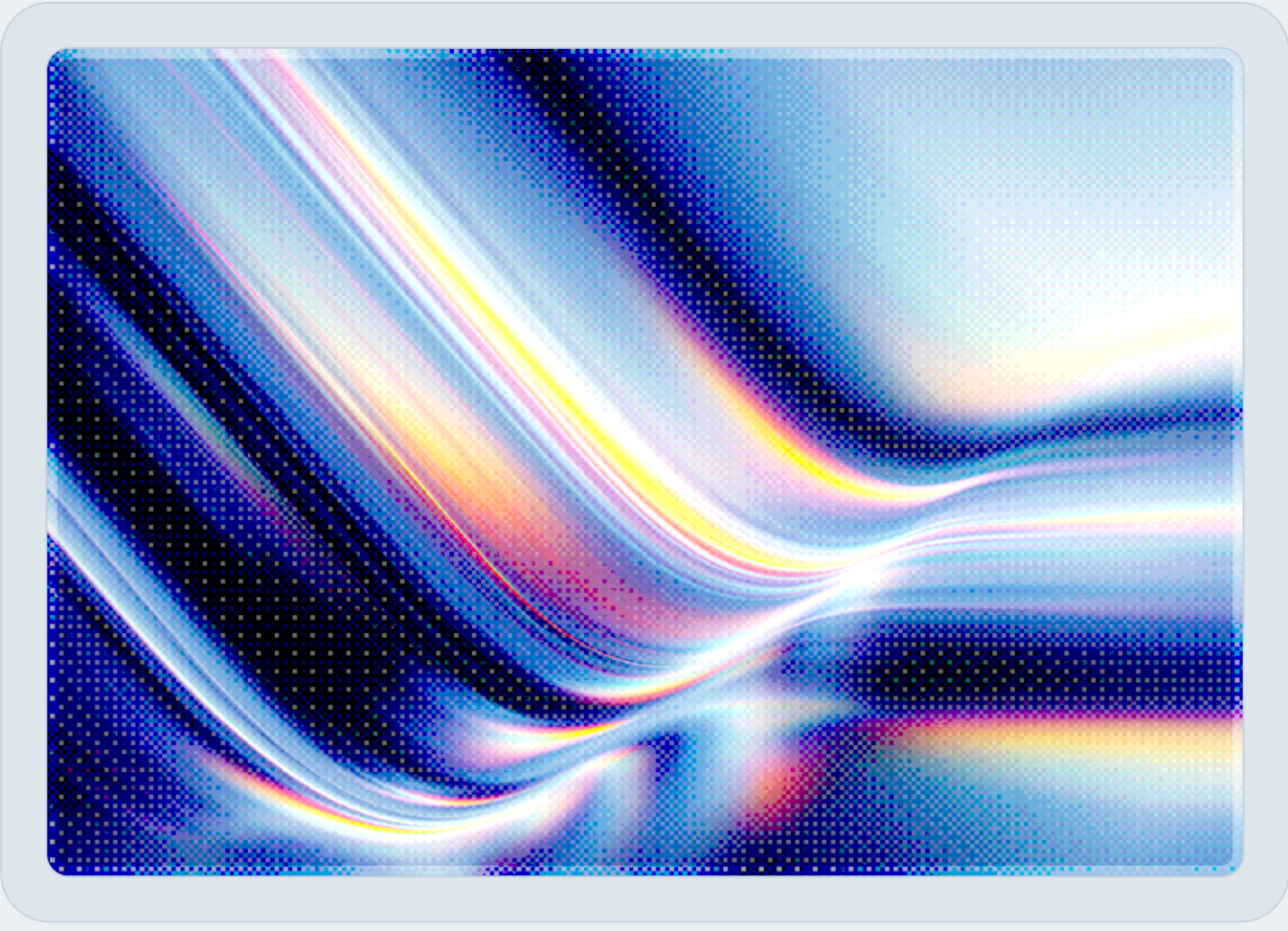
CSAI Foundation | Cloud Security Alliance

NATS-as-C2: Messaging Infrastructure Weaponized for Credential Theft

How Attackers Abuse Cloud-Native Pub/Sub to Exfiltrate AWS and AI API Keys

2026-05-14

AI-assisted Rapid Research



CSAI | CSA cloud security alliance®

Key Takeaways

- The Sysdig Threat Research Team has documented what appears to be the first publicly known use of NATS — a cloud-native, open-source messaging system — as active command-and-control (C2) infrastructure, marking a novel evolution in attacker tradecraft against AI and cloud environments [1].
- The campaign exploited CVE-2026-33017, a critical unauthenticated remote code execution (RCE) vulnerability in Langflow (CVSS 9.3), to gain initial access within twenty hours of public disclosure, demonstrating the rapid weaponization pace that now defines the AI tooling threat landscape [2, 14].
- NATS pub/sub subjects and JetStream durable task queues were used to orchestrate a distributed credential-hunting worker pool, enabling attackers to parallelize harvesting of AWS credentials, OpenAI API keys, Anthropic API keys, and database connection strings across multiple compromised nodes simultaneously [1].
- Eleven CVEs affecting NATS server were publicly disclosed in a single March 2026 patch batch — covering MQTT authentication, JetStream authorization bypass, identity spoofing, and pre-authentication denial-of-service — indicating significant recent attention to NATS security and a broadened attack surface for operators running unpatched deployments [3].
- The LLMjacking threat, of which this campaign is a direct example, has grown 376% in credential theft activity targeting AI services between Q4 2025 and Q1 2026, with reported victim costs reaching six figures per day when flagship models are abused under stolen credentials [4].

Background

NATS: Cloud-Native Messaging at Scale

NATS is a high-performance, open-source messaging system designed for cloud-native and edge deployments. Originally developed by Derek Collison and donated to the Cloud Native Computing Foundation, it is now an incubating project under CNCF governance and is widely used in microservice architectures, IoT platforms, and distributed AI pipelines [7]. The system operates on a publish-subscribe

model: publishers send messages to named subjects, and any connected subscriber with a matching interest receives those messages in near real-time. The companion JetStream engine extends this core model with persistence, durable consumers, and at-least-once delivery — making NATS a capable alternative to Apache Kafka or AWS SQS for teams that prefer a single lightweight binary.

NATS's performance characteristics — sub-millisecond latency, millions of messages per second on commodity hardware, and a small operational footprint — have made it attractive across many legitimate use cases. That same footprint makes it dangerous when left unprotected or when misconfigured deployments are reachable from the internet. By default, a NATS server starts with no authentication or authorization configured. Once a server is exposed on its default port (4222), any client that can reach it may publish, subscribe, and observe traffic unless an operator has explicitly enabled credentials or token authentication [7].

AI Pipeline Tooling: Credential-Rich, Widely Exposed

Langflow is an open-source, visual framework for building AI agent pipelines and retrieval-augmented generation workflows. Like similar tools — LlamaIndex workflows, Flowise, Dify, and others in the space — it is designed to wire together foundation models, vector databases, and external APIs via a graphical interface, storing connection credentials, API keys, and database URIs in configuration files and process environment variables. Organizations use these tools to prototype and deploy AI applications rapidly, often exposing them on network-accessible ports during development with the expectation that they are on trusted networks.

This combination — credentials stored in easily harvested environment variables, endpoints reachable on open ports, and rapid community adoption that outpaces security hardening — has made AI pipeline tools a preferred initial access target. Langflow deployments typically hold API keys for multiple AI providers simultaneously (OpenAI, Anthropic, Google Vertex, AWS Bedrock) alongside the cloud credentials needed for the underlying infrastructure. A single compromised instance can therefore yield a full set of pre-authenticated access tokens across an organization's AI and cloud footprint.

LLMjacking: The Monetization of Stolen AI API Keys

LLMjacking refers to the practice of using stolen AI service credentials to conduct unauthorized inference, effectively running the attacker's workloads against the victim's billing account [6]. Unlike traditional credential theft, where monetization requires selling access or pivoting to financial systems, LLMjacking converts stolen API keys directly into compute — accessible immediately, globally, and without secondary infrastructure. A stolen OpenAI API key has been observed selling for as little as $30 on criminal markets while reportedly enabling victim costs exceeding $46,000 per day from a single

compromised key [6]. In one documented case from March 2026, a developer reportedly received an $82,000 Gemini API bill generated in 48 hours from a single compromised key. Sysdig research tracking this threat documented a 376% increase in credential theft activity targeting AI services between Q4 2025 and Q1 2026 [4]. Operation Bizarre Bazaar, an attacker collective tracked by Pillar Security in early 2026, recorded over 35,000 attack sessions during a campaign spanning December 2025 through January 2026 [5].

Security Analysis

CVE-2026-33017: Zero-Day Exploitation of Langflow

The triggering vulnerability for the campaign described in this note is CVE-2026-33017, an unauthenticated remote code execution flaw in all Langflow versions prior to 1.9.0 [2]. The vulnerable endpoint — `POST /api/v1/build_public_tmp/{flow_id}/flow` — was designed to allow unauthenticated users to build public flows. It accepts attacker-supplied flow data, including arbitrary Python code embedded in node definitions, and executes that code server-side without sandboxing. A single well-crafted HTTP request is sufficient for complete system compromise.

The Sysdig Threat Research Team observed six distinct source IP addresses conducting exploitation attempts within 48 hours of the March 2026 advisory publication, organized into three phases of increasing operational sophistication [2]. No public proof-of-concept code existed at the time of the first exploitation attempts, indicating that at least some actors built functional weaponization directly from the advisory text. This timeline — public disclosure to active exploitation in under twenty hours — is consistent with the rapid-weaponization pattern documented across AI tooling CVEs throughout 2025 and 2026.

Attackers executing via this endpoint ran standard reconnaissance commands (`id`, `ls -al /root`, `cat /etc/passwd`, `env`) to enumerate the host environment and extract sensitive variables from memory. A targeted filesystem sweep (`find /app -name "*.db" -o -name "*.env"`) then located configuration files containing application secrets. The harvested output — API keys for OpenAI, Anthropic, and AWS, along with database connection strings — was base64-encoded and exfiltrated to attacker-controlled callback infrastructure [2, 15].

The NATS-as-C2 Technique

What distinguishes this campaign from prior Langflow exploitation is what happened after initial access. Rather than simply calling home with collected credentials, one operator deployed a stage-two payload that established the compromised Langflow host as a participant in a larger, NATS-coordinated worker pool [1]. This represents, to the knowledge of current public research, the first documented use of NATS as active C2 infrastructure.

The architecture the attackers constructed mirrors legitimate cloud-native distributed work queues. A central NATS server — controlled by the threat actor — acts as the message bus. Compromised hosts subscribe to subjects designated for task distribution, receive credential-hunting instructions via pub/sub messages, execute those instructions locally, and publish results back to collection subjects. JetStream's durable consumer model ensures that work items are not lost if a worker goes offline; incomplete tasks are requeued automatically. From the NATS server's perspective, these are ordinary connected clients exchanging messages on ordinary subjects. From a network monitoring perspective that relies on protocol signatures or known-malicious infrastructure lists, the traffic is difficult to distinguish from legitimate application messaging without behavioral baselining.

This design has several operational advantages for attackers. First, it decouples the attacker's control plane from any individual compromised host — no direct connection from attacker to victim is required after initial payload delivery. Second, NATS's publish-subscribe fan-out means a single instruction can reach hundreds of workers simultaneously. Third, because NATS is a legitimate, widely deployed technology in cloud environments, its traffic is unlikely to trigger alert rules tuned for known C2 protocols. Fourth, if operators have deployed NATS servers with default or weak authentication — a common misconfiguration — attackers can potentially pivot directly onto those servers to access internal messaging infrastructure without deploying any custom tooling.

NATS Vulnerability Landscape: 2025–2026

The NATS project disclosed eleven security advisories in a single batch on March 24, 2026, reflecting both an increased security review effort by the project team and a growing body of discovered issues [3]. The relevant vulnerabilities span several categories that are directly pertinent to the NATS-as-C2 threat model.

Authorization failures in JetStream represent the most operationally consequential cluster. CVE-2026-33222 (CVSS 4.9, Medium) allows a user with JetStream stream restore permissions to restore data to arbitrary stream names outside their intended scope — an integrity bypass that could allow an attacker with limited account access to overwrite or corrupt streams belonging to other tenants [3]. The earlier CVE-2025-30215 (CVSS 9.6, Critical) affected NATS server versions 2.2.0 through 2.11.0-RC.1 and

exposed four JetStream admin APIs — account purge, server remove, account stream move, and account stream cancel-move — to execution by any user with JetStream management permissions, without enforcement of account boundaries [8]. Any deployment still running an unpatched version within that range is fully exposed to cross-tenant data destruction.

Identity and authentication issues present a second significant exposure surface. CVE-2026-33248 documents spoofing of the `Nats-Request-Info` identity header used in leafnode connections, while CVE-2026-33223 identifies the same header as spoofable in the general server context — allowing a client to impersonate the identity of another client or service [3]. CVE-2026-33247 exposes credentials via the server's command-line argument list, readable through the monitoring endpoint, in deployments where process arguments are accessible to monitoring tools [3]. Together, these issues mean that a NATS deployment with outdated software may not only be vulnerable to unauthorized access but may actively leak credentials and allow identity impersonation across the messaging fabric.

MQTT support in NATS introduces additional vectors. CVE-2026-33216 exposes plaintext passwords in MQTT sessions, while CVE-2026-33215 allows session hijacking through Client ID collision [3]. Organizations that have enabled NATS's MQTT bridge as a convenience layer for IoT or legacy protocol compatibility should treat those endpoints as high-priority patching targets.

Why AI Pipelines Are Structurally Vulnerable

The attack described in this note is not an anomaly — it reflects a structural mismatch between the deployment practices typical of AI pipeline tooling and the threat model appropriate for systems that hold credentials with significant financial value. Langflow and similar tools are often deployed using default configurations, which prioritize ease of setup but leave authentication and network exposure controls unconfigured. Environment variable storage of API keys is convenient during development but becomes a readily accessible source of credentials when any component of the pipeline is compromised. The same openness that makes AI pipeline tools easy to integrate makes them easy to exploit.

The financial incentive created by LLMjacking reinforces this targeting. A threat actor who compromises one Langflow instance does not merely gain access to that organization's AI workload — they gain access to pre-funded, immediately usable API credits across every provider whose keys were stored there. In a typical mid-size enterprise deployment, a single compromised Langflow environment might carry credentials for three or four foundation model providers, a cloud provider's IAM keys, and one or more vector database credentials. The attacker's distributed NATS worker pool then amplifies this yield by coordinating systematic harvesting across many compromised instances simultaneously, collecting a portfolio of keys that can be resold, used for LLMjacking operations, or leveraged for lateral movement into connected cloud environments.

Recommendations

Immediate Actions

Patch Langflow immediately. Any deployment running Langflow version 1.8.1 or earlier is vulnerable to CVE-2026-33017. Organizations should upgrade to version 1.9.0 or later without delay. If immediate patching is not operationally possible, restrict network access to Langflow instances using host-based firewall rules or security group policies to allow connections only from trusted IP ranges, and treat any currently exposed instance as potentially compromised pending forensic review.

Audit NATS deployments for authentication configuration. Any NATS server — whether attacker-controlled or legitimately operated — that is reachable from untrusted networks without authentication is an exploitable target. Review running NATS configurations to verify that TLS, NKeys, credentials-based authentication, or token authentication is enabled, and that no implicit `$G` default account remains active for unauthenticated access. Revoke and rotate any credentials passed to the NATS server via command-line arguments or stored in configuration files if the server is running a version affected by CVE-2026-33247, which exposes those credentials via the monitoring endpoint.

Rotate potentially exposed AI API keys and cloud credentials. Any organization that operated a Langflow instance reachable from the internet during the March–May 2026 period should treat all credentials stored in that environment as compromised. This includes API keys for OpenAI, Anthropic, Google, AWS, Azure, and any other provider keys present in `.env` files, application config, or process environment variables. Rotate keys immediately, audit recent API usage logs for anomalous activity patterns indicating unauthorized use, and implement provider-side spending limits and anomaly alerts as an immediate compensating control.

Apply NATS security patches. Deployments running nats-server prior to 2.11.15 or 2.12.6 are affected by one or more of the March 2026 advisories. The critical CVE-2025-30215 requires upgrade to 2.11.1 or later. Operators who previously upgraded to 2.11.1 to address CVE-2025-30215 remain exposed to the March 2026 advisories and must upgrade further to 2.11.15 or 2.12.6. Operators should review the full advisory list at advisories.nats.io and apply the appropriate patched version for their release train [3].

Short-Term Mitigations

Organizations should introduce network-level visibility into NATS traffic patterns within their environments. NATS operates primarily on TCP port 4222 for client connections and 8222 for the monitoring HTTP endpoint. Baseline measurement of normal message rates, subject patterns, and

connected client counts will enable detection of anomalous connections — particularly from recently compromised hosts that have joined an attacker-controlled worker pool. Organizations should alert on any NATS client connections from hosts that do not legitimately participate in messaging workflows.

AI pipeline tooling should be migrated away from environment variable credential storage toward dedicated secrets management infrastructure. Solutions such as HashiCorp Vault, AWS Secrets Manager, Azure Key Vault, or GCP Secret Manager provide short-lived, audited, dynamically rotated credentials that are not present in process memory as plain text. Integration with these systems is supported by most foundation model provider SDKs and should be standard practice for any AI pipeline operating at production scale. Credential injection at runtime — rather than baking keys into configuration files or container images — substantially reduces the harvest value of any single compromised host.

API key usage monitoring with anomaly detection should be deployed at all AI service providers in use. Most major providers now offer spend alerts, per-key usage dashboards, and rate limiting controls that can cap the financial damage from a compromised key. These controls do not prevent theft but do limit the blast radius of successful exfiltration.

Strategic Considerations

The NATS-as-C2 technique may be an early indicator of a potential broader shift: if attackers find that legitimate cloud-native infrastructure provides effective cover, messaging systems, workflow orchestrators, object storage, and event buses become plausible future C2 channels. Security teams should extend their threat models to include these components as potential C2 planes and apply the same access controls and monitoring rigor to them that they apply to traditional network protocols. The key detection challenge is that attacker traffic in this pattern is structurally identical to legitimate application traffic — effective defense requires behavioral baselining of normal usage patterns, not signature-based detection.

AI pipeline environments should receive dedicated threat modeling exercises that account for the credential aggregation function they serve. A single Langflow, Flowise, or similar deployment can accumulate keys spanning an organization's entire AI and cloud provider footprint. The threat model for these environments should treat the credential store as the primary target — not the compute or data the pipeline produces — and design access controls, network segmentation, and detection accordingly.

CSA Resource Alignment

The threats documented in this note map directly to several active Cloud Security Alliance research and framework initiatives.

MAESTRO (Multi-Agent Environment, Security, Threat, Risk, and Outcome) is the CSA's agentic AI threat modeling framework, designed to provide a structured, layer-by-layer method for identifying and mitigating risks in AI agent systems [9]. The NATS-as-C2 technique is particularly relevant to MAESTRO's treatment of infrastructure-layer threats — specifically how the communication fabric beneath AI agent orchestration can be subverted to redirect task dispatch, intercept results, or inject attacker-controlled instructions into legitimate workflows. Organizations building production AI pipelines should apply MAESTRO analysis to their messaging and orchestration layers, not only to the model and agent code. CSA has published application guides for MAESTRO against the OpenAI Responses API and Google A2A protocol [10].

The AI Controls Matrix (AICM) provides a structured set of security controls for AI systems, including domains covering AI supply chain security, secrets and credential management, and AI pipeline access controls [11]. The credential harvesting pattern documented here falls within AICM's Application Provider (AP) control domain, which addresses how organizations that build on top of foundation models should protect the API keys and service bindings that constitute their AI supply chain. AICM auditing guidelines for Application Providers provide actionable mappings from these controls to implementation patterns.

The Cloud Controls Matrix (CCM) addresses the infrastructure and identity controls most directly relevant to NATS hardening and cloud credential protection. CCM domains covering Identity and Access Management (IAM), Infrastructure and Virtualization Security (IVS), and Security Incident Management and Event Reporting (SEF) provide the control framework against which organizations should measure their NATS authentication posture, key rotation practices, and incident response readiness [12].

The CSA blog post "API Security in the AI Era" (September 2025) provides complementary guidance on implementing least-privilege access, short-lived tokens, and API gateway enforcement for AI-driven API environments — controls that directly address the credential exposure pattern exploited in this campaign [13].

References

[1] Sysdig Threat Research Team. "NATS-as-C2: Inside a new technique attackers are using to harvest cloud credentials and AI API keys." Sysdig, May 2026.

[2] Sysdig Threat Research Team. "CVE-2026-33017: How attackers compromised Langflow AI pipelines in 20 hours." Sysdig, March 2026.

[3] NATS Project. "Security Advisories for the NATS.io Project." NATS Project, updated 2026.

[4] Sysdig. "2026 Cloud-Native Security and Usage Report." Sysdig, 2026.

[5] Pillar Security. "Operation Bizarre Bazaar." Pillar Security, 2026.

[6] Sysdig Threat Research Team. "LLMjacking: Stolen Cloud Credentials Used in New AI Attack." Sysdig, 2024.

[7] NATS.io. "NATS – Cloud and Edge Native Messaging." NATS.io, accessed May 2026.

[8] Vulert. "CVE-2025-30215: NATS Server Critical JetStream API Authorization Bypass." Vulert Vulnerability Database, 2025.

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

[10] Cloud Security Alliance. "Threat Modeling OpenAI's Responses API with MAESTRO." CSA Blog, March 2025.

[11] Cloud Security Alliance. "AICM Implementation and Auditing Guidelines (Frameworks)." CSA, 2025.

[12] Cloud Security Alliance. "Cloud Controls Matrix v4.1." CSA, accessed May 2026.

[13] Alex Vakulov. "API Security in the AI Era: Best Practices for AI-Driven APIs." CSA Blog, September 2025.

[14] The Hacker News. "Critical Langflow Flaw CVE-2026-33017 Triggers Attacks within 20 Hours of Disclosure." The Hacker News, March 2026.

[15] MITRE ATT&CK. "Exfiltration Over C2 Channel – T1041." MITRE ATT&CK Enterprise, accessed May 2026.

