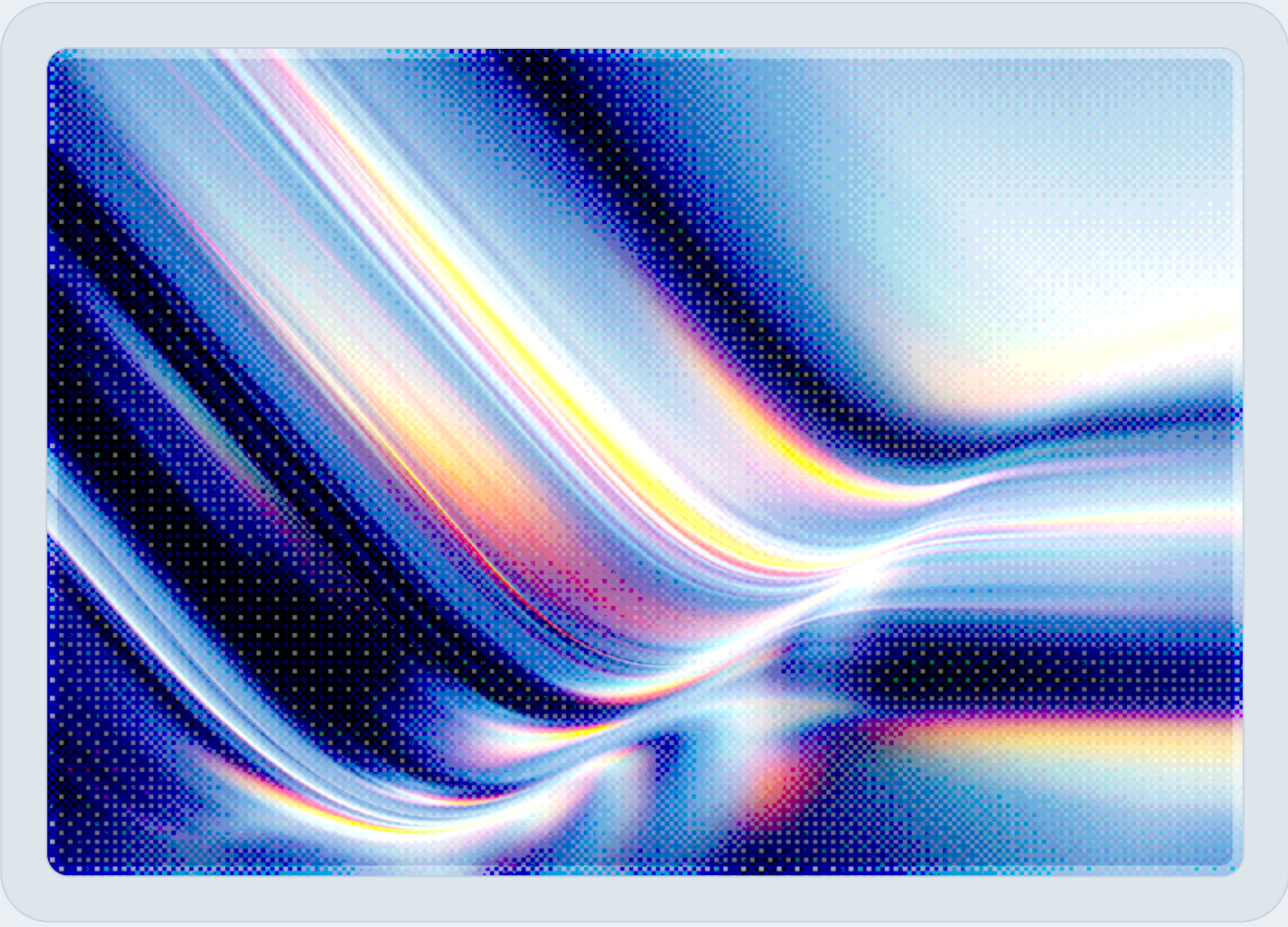
CSAI Foundation | Cloud Security Alliance

RufRoot: Unauthenticated RCE in Ruflo's MCP Bridge

2026-08-05

AI-assisted Rapid Research



CSAI

CSA cloud security alliance®

Key Takeaways

Ruflo, an open-source AI agent orchestration platform built on top of Claude Code and Codex, shipped a default deployment configuration that exposed its Model Context Protocol (MCP) bridge to the open network with no authentication whatsoever. The resulting flaw, tracked as CVE-2026-59726 and nicknamed RufRoot, carries the maximum CVSS base score of 10.0 and allowed any network-reachable attacker to execute arbitrary shell commands, steal LLM provider credentials, and durably poison an organization's AI memory store in ways that survive a software patch, via a single unauthenticated HTTP request [1][2][4]. Noma Security's research arm, Noma Labs, disclosed the vulnerability on June 30, 2026, and Ruflo's maintainer shipped a fix in version 3.16.3 within roughly 24 hours [1][2]. What distinguishes RufRoot from a typical critical RCE is its persistence problem: because the flaw let attackers write directly into Ruflo's AgentDB learning store, applying the patch closes the network entry point but does nothing to remove instructions an attacker already planted in an agent's memory, a characteristic that led Dark Reading to describe the flaw as "patch-resistant" [3]. Organizations that ran any pre-3.16.3 version of Ruflo with internet- or LAN-facing exposure should treat the incident as a full compromise rather than a vulnerability to be closed — a risk-based posture warranted by the flaw's zero-interaction exploitation path, not a claim that every exposed deployment was in fact breached — rotating credentials and auditing stored agent memory rather than simply updating the software.

Background

Ruflo, formerly distributed under the name Claude Flow, is an open-source AI agent orchestration platform designed to coordinate "swarms" of autonomous agents built on Anthropic's Claude Code and OpenAI's Codex [1][2]. Reported adoption figures vary by source and measurement method: The Hacker News cited more than 66,500 GitHub stars for the project, while Noma Labs' own research reported roughly 10 million downloads and approximately 1 million active users, underscoring that this is a widely deployed piece of AI infrastructure rather than a niche tool [1][2]. Ruflo's core function is to let developers stand up multi-agent workflows that share state, coordinate task execution, and persist learned patterns across sessions so that agent behavior improves over time.

That coordination function is implemented through an MCP bridge, a purpose-built Express.js server that translates JSON-RPC requests arriving over HTTP into calls against Ruflo's internal tool library. As of the vulnerable release line, that tool library exposed 233 distinct tools through the bridge, spanning

shell command execution, database read/write operations, agent-swarm spawning, and access to Ruflo's persistent memory subsystem, AgentDB [2]. The project's default `docker-compose.yml` deployment template bound this bridge to port 3001 on all network interfaces (`0.0.0.0`) rather than to the loopback interface, meaning any container or host deployed with the out-of-the-box configuration was reachable from the local network, and, depending on the cloud provider's default network policy, potentially from the public internet as well [1][2]. A companion MongoDB instance used for conversation storage was likewise deployed without authentication enabled by default, compounding the exposure [1].

Security Analysis

The vulnerability's mechanics are straightforward, which is precisely what makes it severe. The MCP bridge's `/mcp` and `/mcp/:group` endpoints implement the standard MCP JSON-RPC protocol and pass any incoming `tools/call` request directly to Ruflo's internal `executeTool()` function with no authentication layer in front of it [2][4]. Ruflo does maintain a command blocklist, `AUTOPILOT_BLOCKED_PATTERNS`, intended to prevent dangerous shell operations, but that control is scoped only to the platform's autopilot workflow; the `/mcp` endpoint bypasses it entirely [2]. The result is that a single unauthenticated HTTP POST request naming the `ruflo__terminal_execute` tool executes arbitrary commands inside the bridge container as the `node` user, with no credentials, session token, or prior interaction required [1][2]. The National Vulnerability Database's entry for CVE-2026-59726 assigns a CVSS 3.1 vector of `AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H` — network-exploitable, low complexity, no privileges or user interaction required, and complete impact to confidentiality, integrity, and availability with a scope change — consistent with the maximum base score of 10.0 [4].

Noma Labs' proof-of-concept demonstrated that this single foothold chains cleanly into full environment compromise. The published attack sequence proceeds through reconnaissance, exploitation, and cleanup phases summarized in Table 1.

Stage	Action	Effect
1. Reconnaissance	Enumerate all 233 exposed tools via `tools/list`	Attacker maps the full capability surface without authentication

Stage	Action	Effect
2. Initial access	Invoke `ruflo__terminal_execute` over the unauthenticated `/mcp` endpoint	Arbitrary shell command execution as the `node` container user
3. Credential theft	Read environment variables from the compromised container	Exposure of LLM provider API keys (Anthropic, OpenAI, etc.)
4. Agent weaponization	Use stolen credentials and orchestration tools to spawn new agent swarms	Attacker-controlled agents operate under the victim's paid API access
5. Memory poisoning	Write malicious entries via `ruflo__agentdb_pattern-store`	Persistent, false "learned" instructions embedded in the AI's long-term memory
6. Data exfiltration	Query the co-located, unauthenticated MongoDB instance on port 27017	Bulk theft of stored agent conversations and workflow history
7. Persistence	Write a beacon script to `/app/beacon.js` and inject a reference into `index.js`	Backdoor survives container restarts via Docker's restart policy
8. Cleanup	Clear shell history on the compromised container	Reduced forensic trail of the intrusion

Table 1. Noma Labs' documented RufRoot exploitation chain [2].

Step 5 is the mechanism behind RufRoot's lingering risk and warrants closer examination. Noma Labs demonstrated that an attacker could plant a fabricated policy in AgentDB — in their proof-of-concept, a fake SOC2 compliance rule — that instructed the AI to silently embed an attacker-controlled URL in every deployment script the system subsequently generated for any user [2]. Because AgentDB is designed to persist and propagate learned patterns precisely so that agent behavior improves across sessions, a poisoned entry behaves exactly as the system intends: it keeps influencing outputs long after the attacker has left the network and long after the underlying software vulnerability has been patched [2][3]. As one industry summary of the incident put it, "updating Ruflo closes the original entry point but

does not remove instructions already written into AgentDB," which means organizations cannot treat patching alone as evidence that the environment is clean [5]. This is the basis for Dark Reading's characterization of RufRoot as a "patch-resistant" flaw capable of leaving "malicious AI agent swarms" operating even after remediation begins [3].

Recommendations

Immediate Actions

Any organization running a self-hosted Ruflo deployment prior to version 3.16.3 should assume compromise rather than wait for evidence of exploitation, given the flaw's zero-interaction, unauthenticated nature. Close inbound access to ports 3001 (MCP bridge) and 27017 (MongoDB) at the firewall immediately, even before upgrading, to stop further exploitation while remediation is planned [1][2]. Upgrade to Ruflo 3.16.3 or later, which changes the default MCP bridge binding to loopback-only, requires bearer-token authentication with constant-time comparison for any remote access, disables the `terminal_execute` tool by default behind an explicit `MCP_ENABLE_TERMINAL` flag, and enforces MongoDB authentication [2]. Because the vulnerability exposed environment variables to arbitrary command execution, rotate every LLM provider API key and any other credential that was reachable from the compromised container, treating all of them as disclosed regardless of whether misuse has been observed [1][2].

Short-Term Mitigations

Patching the software addresses the network-facing vulnerability, but it does not address content an attacker may have already written into persistent storage. Audit the AgentDB pattern store for entries that do not correspond to legitimate, reviewed policies or learned behaviors, paying particular attention to entries that reference external URLs, alter compliance or deployment guidance, or otherwise instruct the agent to take actions a human would not have approved [2][3]. Review MongoDB contents for tampering or unauthorized access, and rebuild affected containers from clean images rather than patching in place, since a persistence mechanism such as the beacon script documented by Noma Labs can survive an in-place software update if the underlying container image and filesystem are not replaced [2]. Organizations should also review outbound network logs from the affected period for connections to unfamiliar hosts, which may indicate either data exfiltration or a beacon checking in with attacker infrastructure.

Strategic Considerations

RufRoot is best understood as a case study in a broader pattern: AI orchestration platforms are shipping powerful, tool-calling network services with the security posture of an internal development convenience rather than a production control plane. The default-open MCP bridge, the scope-limited command blocklist, and the co-located unauthenticated database all have the effect of a security posture suited to local development rather than internet-facing deployment, whether or not that was the maintainers' explicit design intent, yet the software is routinely deployed in exactly that context [1][2]. Security teams evaluating any agent-orchestration or MCP-based platform should require, as a baseline, that all tool-invocation endpoints authenticate by default, that destructive or high-privilege tools (shell execution, database writes, credential access) are disabled unless explicitly and narrowly enabled, and that any persistent memory or pattern store used to shape agent behavior is treated as a sensitive data store subject to integrity monitoring, not merely a cache. The memory-poisoning dimension of this incident also argues for building agent memory review into standard incident response playbooks: a network vulnerability disclosure should now routinely trigger a check of what an attacker may have taught the system, not only what the attacker may have accessed.

CSA Resource Alignment

RufRoot sits squarely within a pattern of unauthenticated MCP and agent-orchestration RCE flaws that CSA's AI Safety Initiative has tracked closely over the preceding months, and this incident should be read alongside that existing body of work rather than in isolation. CSA's research note on the LangGraph RCE Chain [6] documented a structurally similar failure mode in a different agent-orchestration framework: a chain of vulnerabilities in LangGraph's checkpoint persistence layer that escalated from injection to full server takeover on self-hosted deployments. As with RufRoot, the LangGraph chain showed that state and memory persistence layers built to make agents more capable — checkpointing conversation state, storing learned patterns — are also high-value targets precisely because compromising them lets an attacker corrupt what the agent believes to be true, not just what it can access. CSA's AutoJack [7] research note likewise found that an AI agent framework's MCP surface (in that case, AutoGen Studio's WebSocket endpoint) could be reached without authentication and used to achieve host-level code execution, reinforcing that "localhost equals trusted" and "internal service equals authenticated" are false assumptions across the current generation of agent tooling, not defects unique to any single vendor.

The memory-poisoning dimension of RufRoot connects directly to CSA's research note on MCP Tool Poisoning [8], which examined how adversarial content delivered through the Model Context Protocol can manipulate agent behavior without triggering any conventional access-control violation. That research's finding that "a single poisoned tool description can exfiltrate private repository contents"

describes the same underlying risk class as RufRoot's AgentDB pattern poisoning: once an attacker can influence what an agent treats as trusted instruction or learned fact, technical remediation of the access vulnerability does not by itself restore trustworthy behavior. Finally, organizations assessing their exposure to this class of risk should evaluate their agent-orchestration deployments against CSA's AI Controls Matrix (AICM) v1.1 [9], particularly its domains covering identity and access management for non-human/agent identities and threat and vulnerability management, both of which map directly onto the authentication and patch-management failures at the root of RufRoot. Ruflo's orchestration role is the kind of control point those domains are designed to cover: AICM defines an Orchestrated Service Provider role for platforms that integrate and govern models, a category Ruflo fits squarely within, even though the artifact's landing page does not itself spell out a distinct authentication-and-monitoring baseline for that role.

References

[1] The Hacker News. "Ruflo MCP Flaw Lets Unauthenticated Attackers Run Commands and Poison AI Memory." The Hacker News, July 2026.

[2] Noma Security. "RufRoot: The MCP Bridge Vulnerability That Turns Agents Into Rogue Admins (CVE-2026-59726)." Noma Security Blog, July 2026.

[3] Dark Reading. "Patch-Resistant Ruflo Flaw Can Unleash Malicious AI Agent Swarms." Dark Reading, July 2026.

[4] National Vulnerability Database. "CVE-2026-59726 Detail." NIST NVD, July 2026.

[5] Hackread. "CVSS 10.0 RufRoot Flaw Allowed Attackers to Hijack Ruflo Without Logging In." Hackread, July 2026.

[6] Cloud Security Alliance. "LangGraph RCE Chain: Checkpointer Flaw Enables Server Takeover." CSA AI Safety Initiative, June 14, 2026.

[7] Cloud Security Alliance. "AutoJack: AI Browser Agents Enable Host Code Execution." CSA AI Safety Initiative, June 20, 2026.

[8] Cloud Security Alliance. "MCP Tool Poisoning: Adversarial Hijacking of AI Agent Workflows." CSA AI Safety Initiative, July 2, 2026.

[9] Cloud Security Alliance. "AI Controls Matrix (AICM) v1.1." CSA, June 22, 2026.

