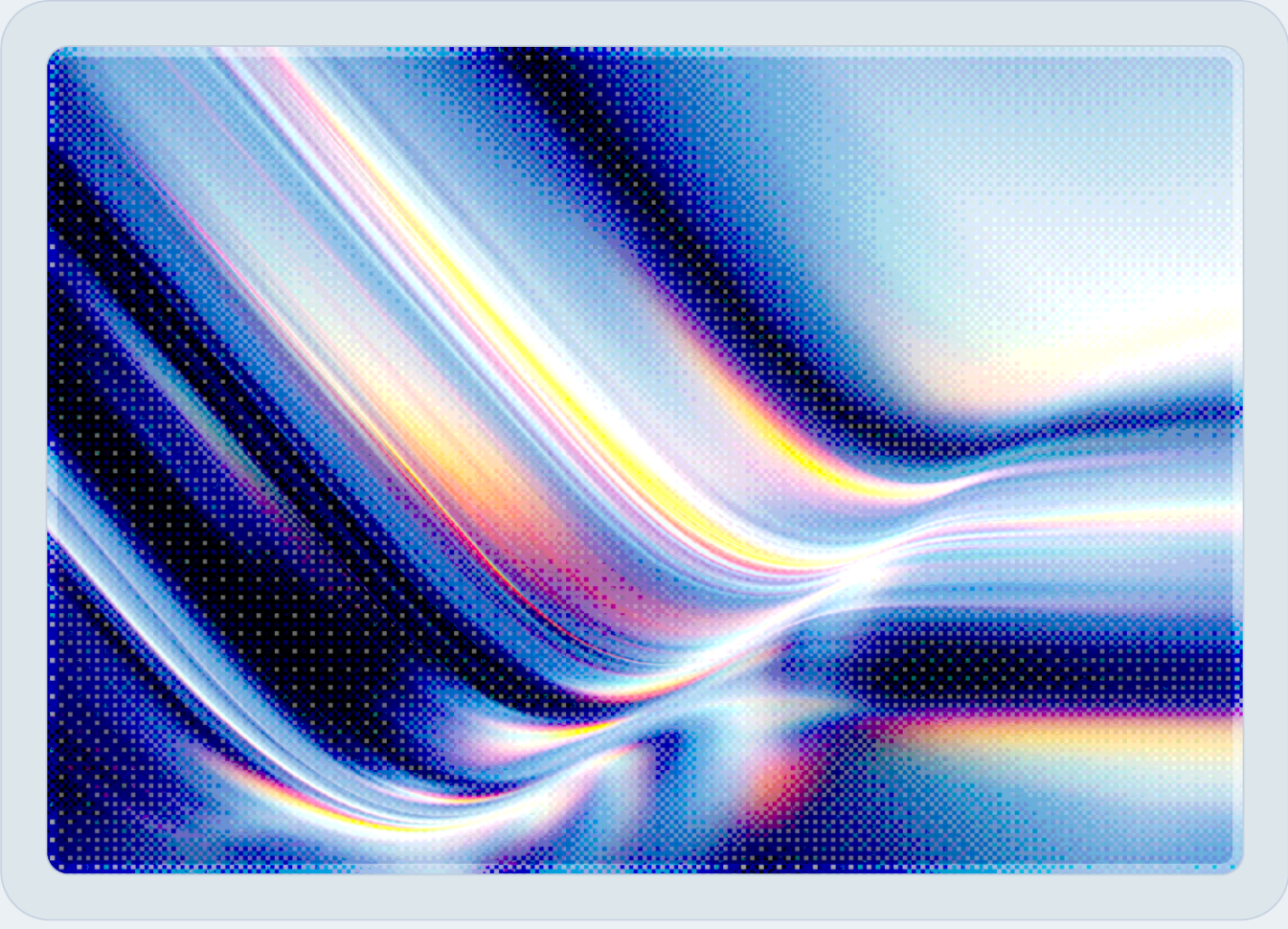
CSAI Foundation | Cloud Security Alliance

CVE-2026-58138: Orkes Conductor RCE Threatens Agentic Workflows

2026-09-20

AI-assisted Rapid Research



CSAI

CSA cloud security alliance®

Key Takeaways

- CVE-2026-58138 is a critical, unauthenticated remote code execution vulnerability (CVSS 3.1: 9.8; CVSS 4.0: 9.3) affecting Orkes Conductor versions 3.21.21 through 3.30.1, patched in version 3.30.2 [1][2][3].
- Attackers can submit crafted workflow definitions containing JavaScript or Python expressions to Conductor's workflow API and achieve arbitrary command execution because the platform's GraalVM script evaluators are configured with unrestricted host access and the API requires no authentication by default [1][2].
- Exploitation is active and recently spiking: FortiGuard Labs recorded roughly 7,000 attack attempts between September 2 and 9, 2026, including a 132 percent day-over-day increase to approximately 1,290 blocked attempts on September 9 alone [1][2][3].
- Conductor is widely deployed as orchestration infrastructure for both conventional microservices and agentic AI workflows [5][6][7], meaning exploitation can reach not just application logic but the coordination layer that governs AI agent behavior, tool access, and human-in-the-loop controls.
- A fix has been available since June 2026, but the CVE identifier was not published until nearly four weeks after the patch shipped, delaying detection by asset-inventory and patch-management tooling that relies on CVE feeds [4].
- This incident extends a pattern CSA has now documented across three AI workflow orchestration platforms in 2026: two separate unauthenticated RCE vulnerabilities in Langflow (CVE-2026-5027 and CVE-2026-33017) and a chained, checkpointer-based RCE in LangGraph each exposed a code-execution path by default [8][11][12]. Three incidents across three platforms in one year do not prove a systemic trend, but they warrant treating unauthenticated code-execution paths as a category-level risk to evaluate in any orchestration engine used for agentic AI.

Background

Orkes Conductor is an open-source workflow orchestration engine, originally developed at Netflix and now maintained and commercially supported by Orkes, that coordinates microservices, APIs, human approval steps, and increasingly AI agents within long-running, stateful workflows. According to the

vendor, Conductor is used by more than 1,200 companies and has over 50,000 developers building on it, with the open-source project maintaining more than 32,000 stars on GitHub [5][6]. Its design goal is to make distributed process execution deterministic, resumable, and auditable, properties that have supported adoption well beyond its original media-streaming use case [5][6].

That same design has extended Conductor's role into the agentic AI stack. Orkes markets Conductor explicitly for building agentic workflows, positioning it as a layer that unifies large language models, retrieval-augmented generation pipelines, tool calls, and human oversight into workflows that can be traced, retried, and governed, and the platform documents support for agent frameworks including LangGraph, the OpenAI Agents SDK, and CrewAI [7]. Enterprises adopting agentic AI have increasingly reached for orchestration engines like Conductor, likely because raw agent frameworks lack the reliability, observability, and access-control primitives that production deployments require. That adoption pattern mirrors what CSA observed earlier in 2026 with Langflow, another AI-oriented workflow platform that suffered an actively exploited, unauthenticated remote code execution vulnerability (CVE-2026-5027), which CSA's analysis attributed in part to rapid adoption that outpaced security hardening [8].

The consequence is that a vulnerability in the orchestration layer extends well beyond a single application. Because Conductor workflows can hold credentials for downstream services, coordinate calls to language models and vector databases, and trigger actions taken by autonomous agents, a code-execution flaw in Conductor itself gives an attacker a foothold inside the control plane that agentic systems depend on for reliability and governance. CVE-2026-58138 illustrates that risk materializing against Conductor specifically.

Security Analysis

CVE-2026-58138 arises from how Conductor evaluates inline scripting expressions embedded in workflow task definitions. Conductor supports task types, including INLINE, LAMBDA, DO_WHILE, and SWITCH, that allow a workflow author to embed JavaScript or Python expressions for conditional logic and data transformation. Those expressions are executed inside a GraalVM polyglot context, and in vulnerable versions that context is configured with `HostAccess.ALL` or `allowAllAccess(true)`, settings that disable the sandboxing GraalVM otherwise provides and expose the full Java host environment to guest-language code [1][2]. An attacker who can submit a workflow definition can therefore write a script expression that escapes the scripting sandbox entirely and invokes operating system commands with the privileges of the Conductor process.

The severity of this flaw is compounded by Conductor's default authentication posture. As SecurityWeek summarized the issue, "no login stands in the way, because the open-source server enforces no authentication by default and leaves its workflow API open" [3]. In practice, this means any internet-exposed Conductor instance running an unpatched version can be compromised by an attacker who never needs valid credentials; submitting a single crafted workflow definition to the public workflow API is sufficient to achieve code execution. The vulnerable version range spans Conductor 3.21.21 through 3.30.1, and the vendor shipped a fix in version 3.30.2 [1][2].

The disclosure and exploitation timeline for CVE-2026-58138 illustrates how quickly opportunistic exploitation followed patch availability, and how a gap in CVE publication can leave patched-but-unlabeled fixes invisible to automated vulnerability management.

Date	Event
June 3, 2026	Orkes ships Conductor 3.30.2, described in release notes only as restricting GraalVM JavaScript further, without a CVE reference [4]
June 30, 2026	CVE-2026-58138 is formally published, nearly four weeks after the fix shipped [4]
Early August 2026	Proof-of-concept exploit code is published to Exploit-DB (EDB-52633) [2]
August 21, 2026	Independent telemetry detects in-the-wild exploitation activity [3][4]
September 2–9, 2026	FortiGuard Labs blocks approximately 7,000 exploitation attempts against Conductor deployments [1]
September 8–9, 2026	Attack volume spikes to roughly 1,290–1,300 blocked attempts in 24 hours, a 132 percent day-over-day increase [1][3]

The gap between the June patch and the June CVE publication matters operationally. Because the fix was described only as a hardening change to GraalVM JavaScript handling rather than as a security advisory, organizations relying on CVE-driven vulnerability scanning had no signal to prioritize the upgrade for nearly a month, and in many cases longer, given that patch adoption typically lags disclosure. As one analysis of the incident put it, "attackers located the open door while the software behind it was still invisible to the people who own it" [4]. FortiGuard's telemetry indicates the attack traffic it observed

originated from a geographically diverse set of source networks, including Germany, Hong Kong, Indonesia, the United Arab Emirates, and India, though source IP geography should be treated as an indicator of attack infrastructure rather than confirmed attacker attribution [1].

The combination of a public, weaponized proof of concept, a default configuration with no authentication, and Conductor's growing footprint in agentic AI deployments creates conditions similar to those CSA flagged in the Langflow advisory: a short window between disclosure and mass scanning, followed by sustained, escalating exploitation against any instance left exposed [8].

Recommendations

Immediate Actions

Organizations running any Orkes Conductor deployment should treat this as an emergency patching event. The single highest-priority action is upgrading every Conductor instance, self-hosted or otherwise, to version 3.30.2 or later, since this closes the unsandboxed evaluator flaw directly. Where immediate upgrade is not feasible, teams should identify and inventory all Conductor deployments across the environment, paying particular attention to instances that may have been provisioned informally to support experimental agentic AI projects and that may not appear in standard asset inventories. Any Conductor workflow API endpoint that is reachable from the public internet should be placed behind network access controls, such as a firewall allowlist, VPN, or reverse proxy requiring authentication, until the upgrade is complete, since the vulnerability is only exploitable by parties who can reach the API.

Short-Term Mitigations

Once patched, organizations should audit Conductor logs for indicators of prior compromise, focusing on unexpected workflow submissions containing INLINE, LAMBDA, DO_WHILE, or SWITCH task definitions with embedded script expressions, as well as any unexplained outbound network connections or process spawning from the Conductor host. Because a successful exploit grants command execution with the privileges of the Conductor process, credentials, API keys, and secrets accessible to that process, including any LLM provider keys, database credentials, or service-to-service tokens used by downstream agents, should be rotated as a precaution if internet exposure cannot be ruled out for the vulnerable window. Deploying Conductor with network micro-segmentation, running the process under a

non-privileged account, and disabling unrestricted GraalVM host access wherever the deployment's workflow logic does not genuinely require it will reduce the blast radius of any future evaluator-related flaw in this or similar platforms.

Strategic Considerations

CVE-2026-58138 is the third major unauthenticated, or near-unauthenticated, remote code execution vulnerability disclosed in 2026 against a workflow orchestration platform used to build agentic AI systems, following two separate Langflow incidents (CVE-2026-5027 and CVE-2026-33017) and a checkpointer-based RCE chain disclosed in LangGraph [8][11][12]. Security teams should treat this as evidence of a pattern rather than an isolated incident: platforms that evaluate user- or workflow-supplied code as a core feature, whether for conditional branching, data transformation, or agent tool invocation, need to be assessed for sandbox integrity and default authentication posture before they are trusted with production agentic workloads. Threat modeling for agentic AI deployments should explicitly evaluate the orchestration and evaluator layer as a distinct attack surface, not merely the language model or the agent's tool permissions, since compromise at the orchestration layer can subvert many of the controls built on top of it. Vulnerability management programs should also account for the disclosure gap observed in this case, where a security-relevant fix shipped without a corresponding CVE for several weeks, by tracking vendor release notes for orchestration and AI infrastructure dependencies directly rather than relying solely on CVE feed ingestion.

CSA Resource Alignment

This incident closely parallels the vulnerability CSA examined in its research note on the Langflow path traversal flaw, and that note remains the most directly applicable prior CSA work. "CVE-2026-5027: Langflow Path Traversal to Unauthenticated RCE" analyzes an actively exploited, unauthenticated code-execution vulnerability in another widely adopted AI workflow platform and provides emergency response, hardening, and threat-hunting guidance that applies with minimal modification to the Conductor incident, including the emphasis on disabling insecure default configurations, restricting network exposure, and rotating credentials after suspected compromise [8]. Security teams responding to CVE-2026-58138 should treat that note as a companion playbook.

Two additional CSA research notes reinforce the same lesson from adjacent platforms. "Langflow RCE CVE-2026-33017: Exploited Within 20 Hours" documents a second, separate unauthenticated RCE in Langflow, and "LangGraph RCE Chain: Checkpointer Flaw Enables Server Takeover" documents a chained vulnerability in LangGraph's state-persistence layer that likewise granted unauthenticated

attackers server-level control [11][12]. Read together with the Conductor incident and the original Langflow note, these four notes span three distinct orchestration platforms within a single year, which is the evidentiary basis for treating unauthenticated code-execution paths in agentic AI infrastructure as a recurring category risk rather than a set of unrelated one-off bugs.

Because Conductor is increasingly deployed as the coordination layer for AI agents rather than purely for conventional microservices, this incident also falls squarely within the scope of CSA's MAESTRO framework for agentic AI threat modeling. MAESTRO's layered reference architecture treats deployment and infrastructure, including the orchestration engines that sequence agent actions and tool calls, as a distinct threat surface requiring its own control analysis, separate from the foundation model or agent framework layers [9]. Organizations building agentic systems on Conductor or comparable orchestration engines should use MAESTRO to evaluate how a compromise at the orchestration layer could propagate into agent behavior, tool misuse, or unauthorized data access.

Finally, the control gaps exposed by this vulnerability, namely the absence of default authentication on a workflow API and the lack of sandboxing around user-supplied code execution, map to domains within CSA's AI Controls Matrix (AICM) v1.1, particularly those covering identity and access management and application and interface security for AI-enabled systems. Organizations conducting vendor risk assessments of orchestration platforms used in AI pipelines can use AICM v1.1 to structure due diligence questions about default authentication posture, code execution sandboxing, and patch disclosure practices before adopting a platform for production agentic workloads [10].

References

[1] The Hacker News. "Critical Pre-Auth RCE in Orkes Conductor Workflow Platform Exploited in the Wild." The Hacker News, September 2026.

[2] FortiGuard Labs. "Orkes Conductor Evaluator Remote Code Execution." Fortinet Outbreak Alert, 2026.

[3] SecurityWeek. "Critical Orkes Conductor Vulnerability Exploited in Attacks." SecurityWeek, September 2026.

[4] Empirical Security. "September 2026 CVE of the Month: The 9.8 Nobody Knows They Are Running (CVE-2026-58138)." Empirical Security Research, September 2026.

[5] Orkes. "Platform – One Platform for Agents and Workflows." Orkes, 2026.

[6] GitHub. "conductor-oss/conductor." Conductor OSS, 2026.

[7] Orkes. "Conductor Agents." Orkes Developer Guide, 2026.

[8] Cloud Security Alliance. "CVE-2026-5027: Langflow Path Traversal to Unauthenticated RCE." CSA AI Safety Initiative, June 2026.

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

[10] Cloud Security Alliance. "AI Controls Matrix (AICM) v1.1." Cloud Security Alliance, 2026.

[11] Cloud Security Alliance. "Langflow RCE CVE-2026-33017: Exploited Within 20 Hours." CSA AI Safety Initiative, 2026.

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

