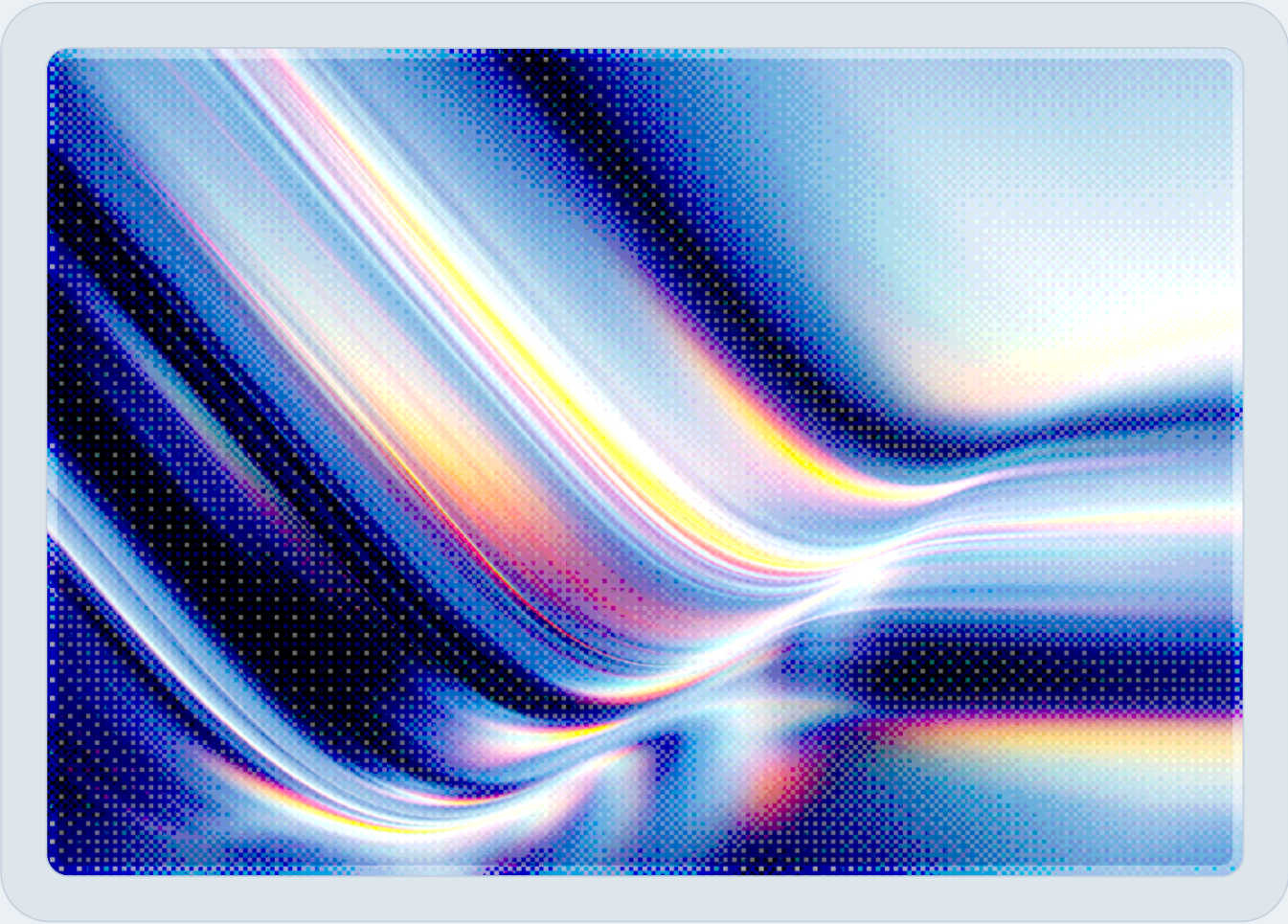
CSAI Foundation | Cloud Security Alliance

MLflow SSRF Under Active Attack: A Platform Breach Vector

2026-08-23

AI-assisted Rapid Research



CSAI

CSA cloud security alliance®

Key Takeaways

Attackers began exploiting an unauthenticated server-side request forgery (SSRF) vulnerability in MLflow, the widely deployed open-source machine learning lifecycle platform, within hours of its CVE assignment, using it to reach cloud metadata endpoints and steal cloud provider credentials [1][2]. The flaw, tracked as CVE-2026-64849 with a CVSS 3.1 score of 9.3, lives in MLflow's webhook-testing functionality and affects all versions prior to 3.15.0 [5][6]. On August 19, 2026, CISA added the vulnerability to its Known Exploited Vulnerabilities (KEV) catalog, requiring federal civilian agencies to remediate by September 2, 2026 [4]. The incident extends a pattern this research program has tracked through 2026 in which AI development and MLOps platforms — rather than the models they train or serve — become the initial-access foothold for cloud environment compromise [7]. Organizations running MLflow Tracking Servers, whether self-hosted or embedded inside a managed AI platform, should treat this as an emergency patching event and should audit cloud identity scopes granted to any workload running MLflow; this program's read of the incident is that the blast radius of this SSRF is determined less by the bug itself than by what the underlying compute identity can do.

Background

MLflow began as an internal Databricks project and was contributed to the Linux Foundation in 2020 to serve as a vendor-neutral, open governance standard for managing the machine learning lifecycle: experiment tracking, model packaging, a model registry, and deployment [8]. Its ubiquity is part of what makes it an attractive target: a flaw here reaches far more environments than a niche tool would, because the platform was already downloaded more than two million times per month by 2020, when it joined the Linux Foundation, and its footprint has only grown since [8]. It is embedded, often invisibly, inside managed offerings from major cloud providers and inside internal data science platforms across the industry, which means a flaw in MLflow itself does not stay confined to a niche tool — it surfaces wherever organizations track experiments or register models.

The vulnerable feature is MLflow's model-registry webhook system, which lets administrators configure the Tracking Server to send HTTP callbacks when registry events occur, and includes a "test" function so a webhook configuration can be validated before it goes live. The relevant endpoint, `POST /api/2.0/mlflow/webhooks/{id}/test`, ships without authentication by default on a

standard MLflow deployment, meaning any network client that can reach the Tracking Server can trigger a webhook delivery and observe its response [5][6]. MLflow's own SSRF protection, a function called `_validate_webhook_url`, checks the destination hostname at configuration time and rejects addresses that resolve to non-public IP ranges. The flaw is that this validation happens only once, against the original URL, while the actual delivery logic in `mlflow/webhooks/delivery.py` follows HTTP redirects without re-validating or pinning the destination address [6]. An attacker can therefore host a public HTTPS endpoint that passes the initial check, then respond with an HTTP 302 redirect to an internal address, an approach that also opens the door to DNS-rebinding variants of the same bypass [5][6]. MLflow's maintainers fixed the issue in version 3.15.0 [5][6].

This class of validate-then-redirect SSRF bypass is a known and recurring anti-pattern, and its consequences are compounded in cloud-native ML infrastructure because Tracking Servers are commonly deployed with broad service identities attached — a pattern this research program has observed in at least one prior comparable case, in which an over-scoped compute identity attached to an internet-reachable service was the central risk driver behind an SSRF-to-credential-theft chain [7]. Once an attacker can make the Tracking Server issue an arbitrary internal HTTP request, the highest-value target is almost always the cloud provider's instance metadata service: AWS's IMDS endpoint at `169.254.169.254`, Google Cloud's `metadata.google.internal`, or Azure's equivalent Instance Metadata Service. These services return temporary identity tokens scoped to whatever role or service account the compute instance carries, and because the `/test` endpoint echoes back the upstream response status and body, an attacker reading that response gets the credential material directly, with no further exploitation steps needed [5][6].

Security Analysis

Exploitation of CVE-2026-64849 began within hours of CVE assignment, a timeline consistent with the fastest exploitation windows this program has tracked in 2026 [1][3]. Independent honeypot telemetry from watchTowr's Attacker Eye network detected scanning and exploitation attempts within hours of the CVE being publicly assigned, and multiple outlets reported automated internet-wide scans for exposed MLflow Tracking Servers within days, though neither outlet attributed the activity to a specific actor type [1][3]. The Hacker News and SecurityWeek both reported that the observed objective was consistent across campaigns: reach the cloud metadata endpoint reachable from the Tracking Server's network position, extract the temporary credential or token issued to that workload's identity, and use it to pivot into the surrounding cloud account [1][3]. BleepingComputer's coverage of CISA's KEV addition confirmed theft of AWS IAM credentials via the metadata service; because the underlying SSRF reaches whatever metadata endpoint is reachable from the Tracking Server's network position, the same

technique would extract Google Cloud service account tokens or Azure Managed Identity tokens on deployments running in those environments, though BleepingComputer's reporting itself was limited to the AWS case [2]. Any of these credential types can grant an attacker access to storage buckets, secrets managers, container registries, or additional compute resources depending on how permissively the underlying identity was scoped.

What makes this incident notable for AI security practitioners specifically, rather than a generic web application SSRF story, is the deployment context. By analogy with a comparable case this program documented in July 2026, MLflow Tracking Servers are plausibly provisioned the same way many quickly-deployed AI platform components are: outside formal security review, and attached to broader-than-necessary compute identities [7]. This report has not independently surveyed MLflow deployment practices, but the deployment pattern observed in that July 2026 review is a documented precedent for the same failure mode recurring here. That review examined how Wiz's autonomous "Red Agent" discovered and exploited an SSRF vulnerability in a production Google Cloud Run service, iteratively probing the service's input validation logic across dozens of interactions until it constructed a request that bypassed validation and reached the internal cloud metadata endpoint — the same well-documented SSRF outcome that enables credential theft against cloud workloads generally [7]. The MLflow case differs in that the vulnerability was found and weaponized by human threat actors rather than an autonomous agent, but the underlying architectural failure mode — validation logic that can be bypassed by a redirect, paired with a metadata service that is reachable by default — is the same, which is at least consistent with a systemic pattern in how cloud-native ML and AI platforms are deployed rather than a one-off coding mistake in a single project. This program has documented two such cases to date, however, and a broader claim of systemic risk would benefit from tracking additional AI/MLOps CVEs as they accumulate through 2026; it remains possible that this reflects the general immaturity of fast-growing open-source tooling rather than something specific to AI infrastructure.

The CISA KEV addition also sits inside a broader regulatory trend this program has tracked since mid-2026: CISA's Binding Operational Directive 26-04, issued June 10, 2026, reprioritizes federal patching timelines using a four-variable risk model — public exposure, exploitation status, automatability, and technical impact — producing tiered remediation deadlines of three, fourteen, or sixty days depending on how many criteria a given vulnerability meets [9]. MLflow's KEV listing on August 19, 2026, carrying a fourteen-day remediation deadline of September 2, 2026 for federal civilian agencies, fits within that framework, and it is a reasonable expectation that as AI and ML tooling continues to proliferate inside both public and private sector environments, CISA will continue to treat exploited vulnerabilities in this software category as warranting expedited handling under BOD 26-04's risk-based approach [4][9].

Recommendations

Immediate Actions

Organizations operating any MLflow Tracking Server, whether self-managed or embedded in a vendor's managed AI platform, should upgrade to MLflow 3.15.0 or later immediately, treating this as an emergency patch rather than a routine update given confirmed active exploitation [5][6]. Security teams should simultaneously inventory every MLflow deployment across the environment, including instances spun up ad hoc by data science or ML engineering teams that may not be registered in the central asset inventory, since unauthenticated, internet-reachable Tracking Servers are the population attackers are actively scanning for [1][3]. Any organization that finds an exposed, unpatched instance should assume compromise and begin credential rotation for the cloud identity attached to that workload, along with a review of API activity logs for the affected account for signs of unauthorized resource access following any period of exposure [2][3].

Short-Term Mitigations

Beyond patching, organizations should audit the cloud identity, whether an IAM role, service account, or managed identity, attached to every MLflow deployment and replace default or broad compute-scope identities with workload-specific, least-privilege credentials, so that a future SSRF or similar flaw in this or any adjacent tool yields a much smaller blast radius [7]. Network-level controls that block outbound access to the cloud metadata service from workloads that do not require it, or that require metadata requests to pass through an egress proxy enforcing the provider's hardened metadata protocols (such as AWS IMDSv2 session tokens), are a defense-in-depth measure this program has recommended in comparable SSRF cases and would reduce exploitability even against redirect- or DNS-rebinding-based bypasses not yet discovered [7]. Tracking Servers that must remain internet-reachable for legitimate business reasons should sit behind authentication at the network or application layer, since the underlying flaw specifically depends on the webhook-test endpoint being reachable without credentials [5][6].

Strategic Considerations

This incident is best understood as one more data point in a pattern rather than an isolated event. AI development, orchestration, and MLOps platforms make an attractive initial-access surface: they are deployed quickly, iterated on rapidly, and frequently granted more cloud privilege than the task requires, and the two incidents this program has now documented (the Cloud Run SSRF case analyzed in July

2026, and the MLflow SSRF case here) are consistent with attackers exploiting that combination, though two incidents are not yet enough to establish it as a deliberate targeting strategy rather than a byproduct of how quickly this tooling category is deployed. Security leaders should incorporate MLflow, and comparable platforms such as Langflow, Kubeflow, and similar orchestration tools, into the same vulnerability management and asset inventory processes applied to customer-facing production services, rather than treating them as internal developer tooling exempt from that rigor. Given that CISA has now added exploited AI/MLOps platform vulnerabilities to its KEV catalog under BOD 26-04's risk-based framework, organizations preparing for similar patching expectations should build the capability to detect and remediate AI/MLOps infrastructure vulnerabilities on the same expedited timelines increasingly being applied to traditional enterprise software [4][9].

CSA Resource Alignment

This incident closely mirrors the exploit chain examined in CSA's Autonomous AI Red Teams: Security Implications and Guidance, which analyzed a case in which an SSRF vulnerability in a cloud-hosted service was chained into cloud metadata access. That note's recommendations, including replacing default compute service accounts with workload-specific least-privilege identities, blocking metadata service access at the network layer where not required, and treating resolved-IP-range allowlists as more robust than domain allowlists, apply directly to MLflow deployments and should form the basis of the mitigation work described above.

CSA's CISA BOD 26-04: AI Threat Forces 3-Day Critical Patch Mandate examined the mechanics of CISA's Binding Operational Directive 26-04 — its four-variable risk matrix and tiered three-, fourteen-, and sixty-day remediation windows, illustrated through the directive's first enforcement instance against an Ivanti Sentry vulnerability. While that note does not address MLflow or Langflow specifically, its analysis of how the directive's risk-based tiers are applied is directly useful for interpreting where MLflow's fourteen-day KEV deadline falls within that framework and for anticipating how future AI platform vulnerabilities are likely to be prioritized under the same directive.

CSA's Langflow Path Traversal: Unauthenticated RCE Actively Exploited, covering CVE-2026-5027, documents a comparable precedent in which a widely deployed, internet-exposed AI development platform shipped an unauthenticated file-write endpoint; in that case active exploitation was first detected 73 days after public disclosure rather than within hours, a reminder that unpatched exposure windows for AI development tooling can persist for months, not just days. Its guidance on inventorying shadow Langflow deployments and hardening internet-exposed AI tooling generalizes well to the MLflow case documented here.

Where these incident-specific artifacts do not cover a control area, the AI Controls Matrix (AICM v1.1) provides the underlying governance baseline, particularly its Threat and Vulnerability Management and Application and Interface Security domains, which map directly to the patch management, identity scoping, and network exposure controls this note recommends [10].

References

[1] The Hacker News. "Attackers Exploit MLflow SSRF Flaw to Steal Cloud Credentials and Secrets." The Hacker News, August 18, 2026.

[2] BleepingComputer. "CISA warns of hackers exploiting critical MLflow vulnerability." BleepingComputer, August 2026.

[3] SecurityWeek. "MLflow Vulnerability Exploited for Cloud Credential Theft." SecurityWeek, August 2026.

[4] CISA. "CISA Adds One Known Exploited Vulnerability to Catalog." Cybersecurity and Infrastructure Security Agency, August 19, 2026.

[5] GitHub Security Advisory Database. "MLflow: Unauthenticated full-read SSRF in webhook delivery – GHSA-7gwp-5pfp-969j." GitHub, 2026.

[6] IONIX Threat Center. "CVE-2026-64849 – Unauthenticated SSRF via Webhook Redirect Bypass – MLflow < 3.15.0." IONIX, 2026.

[7] Cloud Security Alliance. "Autonomous AI Red Teams: Security Implications and Guidance." Cloud Security Alliance, July 10, 2026.

[8] Linux Foundation. "The MLflow Project Joins Linux Foundation." Linux Foundation, June 25, 2020.

[9] CISA. "BOD 26-04: Prioritizing Security Updates Based on Risk." Cybersecurity and Infrastructure Security Agency, June 10, 2026.

[10] Cloud Security Alliance. "AI Controls Matrix (AICM) v1.1." Cloud Security Alliance, 2025.

