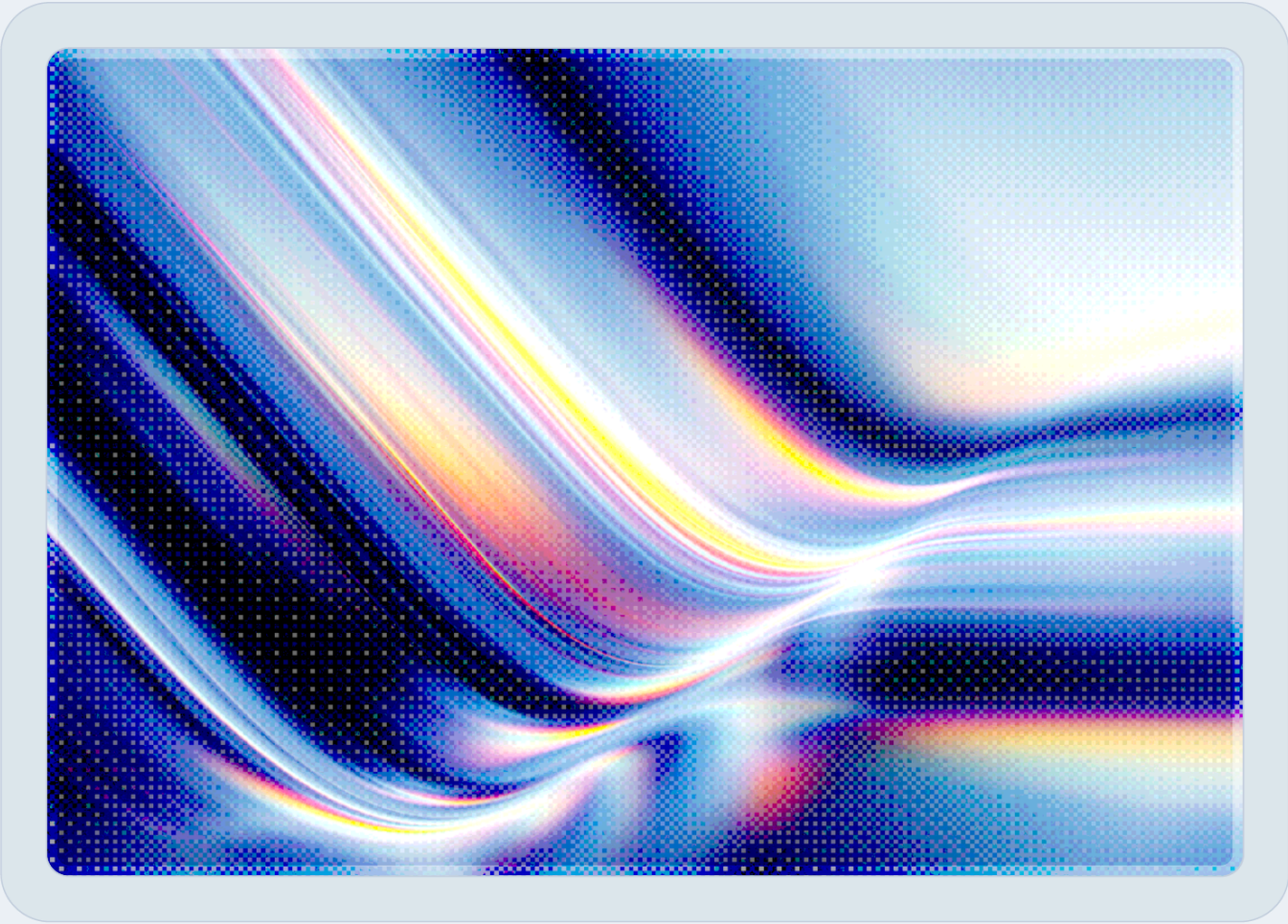
CSAI Foundation | Cloud Security Alliance

MLflow SSRF Actively Exploited for Cloud Credential Theft

CVE-2026-64849: A Redirect-Bypass SSRF in Webhook Delivery

2026-08-26

AI-assisted Rapid Research



CSAI

CSA cloud security alliance®

Key Takeaways

- CVE-2026-64849 is a critical (CVSS v3.1: 9.3) unauthenticated server-side request forgery (SSRF) vulnerability in MLflow's webhook test endpoint, affecting all MLflow versions prior to 3.15.0. [1]
- The root cause is a time-of-check/time-of-use (TOCTOU) gap: MLflow validates a webhook's destination hostname once, then re-resolves it independently when the actual HTTP request is sent, allowing a DNS-rebinding attacker to swap a benign public IP for a cloud metadata address between the two lookups. [2]
- Because the vulnerable `/api/2.0/mlflow/webhooks/{id}/test` endpoint reflects the upstream response back to the caller, a successful attack yields a full, readable SSRF capable of exfiltrating AWS, GCP, or Azure instance-metadata credentials directly to the attacker. [1][3]
- watchTowr Intel's Attacker Eye honeypot network observed threat actors scanning for and exploiting internet-exposed MLflow Tracking Servers within hours of the CVE's public assignment on August 17, 2026, seeking cloud IAM tokens and service-account keys; Rescana separately reported that VulnCheck made similar observations. [4][5]
- CISA added CVE-2026-64849 to its Known Exploited Vulnerabilities catalog on August 19, 2026, requiring Federal Civilian Executive Branch agencies to remediate by September 2, 2026. [6]
- Organizations should upgrade to MLflow 3.15.0 or later immediately. [3] In the interim, teams should block MLflow Tracking Server egress to cloud metadata endpoints [7] and audit webhook configurations for unauthorized entries as a general precaution.

Background

MLflow is an open-source platform for managing the machine learning lifecycle — experiment tracking, model packaging, a model registry, and deployment — and it has become one of the most widely deployed MLOps tools in enterprise and cloud-native AI environments. Its Tracking Server component is frequently run as a shared, internally reachable service that data science and ML engineering teams use to log experiments, register models, and coordinate deployment pipelines. Because that server typically

runs inside a cloud virtual machine or container with an attached IAM role, it inherits the same trust relationship with the cloud provider's instance-metadata service that any other workload on that host would have.

MLflow's model registry supports webhooks that notify external systems when registry events occur, such as a model transitioning between lifecycle stages. To create and test a webhook, MLflow validates the destination URL to prevent it from pointing at internal or link-local addresses — a standard SSRF defense. The flaw disclosed as CVE-2026-64849 defeats that defense: the validation step and the request-execution step resolve the destination hostname independently, and an attacker who controls DNS for the webhook's domain can return a different IP address for each resolution. [2] This class of bug, commonly called DNS rebinding, is a well-understood technique for bypassing hostname-based SSRF filters, and its reappearance in MLflow's webhook subsystem illustrates how easily URL-validation logic can be undermined once redirects or repeated DNS lookups are involved.

The vulnerability was assigned CVE-2026-64849 on August 17, 2026, with a GitHub Security Advisory (GHSA-7gwp-5pfp-969j) crediting the MLflow maintainers with the fix released in version 3.15.0. [1][2] Exploitation began almost immediately. watchTowr Intel reported that its global honeypot network, Attacker Eye, observed attackers "indiscriminately scanning for exposed MLflow instances online within hours of the CVE being assigned," with payloads targeting AWS, Google Cloud, and Azure metadata endpoints. [4] Rescana reported that VulnCheck observed similar scanning activity targeting MLflow Tracking Servers, and CISA's rapid addition of the flaw to its Known Exploited Vulnerabilities catalog on August 19, 2026 — two days after CVE assignment — reflects the urgency federal authorities placed on the threat. [5][6]

Security Analysis

The TOCTOU Redirect Bypass

The vulnerable code path lives in two places: `mlflow/utils/validation.py`, which implements `_validate_webhook_url()`, and `mlflow/webhooks/delivery.py`, which implements the function that actually sends the HTTP request. The validation function resolves the webhook's hostname via DNS and confirms the resolved address is a public, non-internal IP — a reasonable first line of defense against SSRF. Critically, however, it discards that resolved address once the check passes. When the delivery function subsequently sends the actual HTTP request, it resolves the same hostname again, independently, and it is that second resolution — not the one that was validated — which determines where the request actually goes. [2]

An attacker exploiting this gap needs only to control a DNS domain and configure it to answer differently depending on request timing or count: a public IP address on the first (validation) lookup, and a private or link-local address — most consequentially, `169.254.169.254`, the address used by AWS, GCP, and Azure metadata services — on the second (delivery) lookup. [1][2] Because MLflow's webhook test endpoint is unauthenticated and reflects the upstream response status and body directly back to the caller, the attacker does not need any other foothold on the network; a single crafted webhook-test request against an internet-exposed MLflow Tracking Server can return the full contents of the cloud metadata response, including short-lived IAM credentials, directly in the HTTP response. [1][3] The GitHub issue tracking the flaw additionally notes that MLflow's webhook creation endpoint lacked adequate permission restrictions, meaning any authenticated low-privilege user — not just an administrator — could register a malicious webhook, extending the risk to internal, multi-tenant MLflow deployments as well as fully unauthenticated internet-facing ones. [2]

Observed Exploitation and Attacker Objectives

Reporting from watchTowr Intel and Rescana converges on a consistent picture: attackers used mass internet scanning — via services such as Shodan and Censys — to identify exposed MLflow Tracking Servers, then issued SSRF payloads targeting AWS, GCP, and Azure metadata endpoints to harvest IAM access tokens, service-account keys, and Azure Managed Identity tokens; Rescana additionally reported that VulnCheck made corroborating observations. [4][5] The objective in observed cases was credential theft for downstream monetization, lateral movement into the victim's cloud environment, or resale of harvested credentials, rather than direct compromise of the MLflow application itself. [4] This pattern is consistent with SSRF-to-metadata-theft chains generally, in which the application server functions as a stepping stone to the cloud identity fabric surrounding it rather than the attacker's actual objective.

The speed of exploitation — honeypot detections within hours of CVE assignment — reflects both the maturity of automated vulnerability-scanning infrastructure available to opportunistic attackers and the value that cloud credentials extracted via metadata-service SSRF now command. This is consistent with the Langflow CVE-2026-5027 case discussed later in this note, and suggests a pattern worth monitoring across AI-platform disclosures: internet-exposed AI infrastructure may be scanned and weaponized on a timescale of hours, not days, once a CVE becomes public, leaving very little margin for organizations that rely on standard patch-management cadences.

The Fix: Connection-Time Validation

MLflow's remediation, delivered in version 3.15.0 via pull request #24258, closes the TOCTOU window by moving IP validation from the URL-checking stage to the moment the TCP connection is actually established. The fix introduces an `SSRFProtectedHTTPAdapter` in

`mlflow/webhooks/ssrf.py` that inspects the connected socket's peer IP address immediately after the TCP handshake completes — before any TLS negotiation or HTTP data is exchanged — and rejects the connection if that address is not public. [3] Because the check happens against the actual socket the data will flow over, rather than against a hostname that could resolve differently on a later lookup, the fix eliminates the redirect and DNS-rebinding bypass entirely, while still validating the original hostname's certificate for HTTPS connections. The protection is scoped to the webhook-delivery HTTP session specifically, avoiding process-wide changes to MLflow's networking behavior, and it fails closed by raising a dedicated `SSRFProtectionError` rather than allowing automatic retry logic to mask the block. [3] This connection-time validation pattern — checking the socket, not the hostname string — reflects the architecture generally recommended for defending against SSRF in any application that must fetch attacker-influenceable URLs, and organizations building similar webhook or callback features in their own AI tooling should treat it as a reference implementation.

Recommendations

Immediate Actions

Organizations running any self-hosted MLflow Tracking Server should upgrade to version 3.15.0 or later without delay; this is the only complete remediation, since it addresses the underlying validation architecture rather than adding a point patch. [1][3] Before or during the upgrade window, teams should audit all configured webhooks in their MLflow model registries for unrecognized or suspicious destination URLs, since the vulnerability has been exploitable since before public disclosure and any webhook not created by a known, trusted process should be treated as a possible indicator of prior exploitation. Federal agencies and organizations that track CISA KEV deadlines should note the September 2, 2026 remediation date and prioritize accordingly. [6]

Teams should also review cloud audit logs — AWS CloudTrail, Google Cloud Audit Logs, or Azure Activity Log — for anomalous use of any IAM role or managed identity attached to hosts running MLflow, particularly API calls originating from unfamiliar source IPs or unusual service enumeration patterns shortly after any unpatched MLflow instance was internet-reachable. Where such activity is found, or cannot be ruled out, credentials associated with the MLflow host's identity should be rotated and any temporary security tokens invalidated.

Short-Term Mitigations

Where immediate patching is not feasible, organizations should block network egress from MLflow Tracking Server hosts to the cloud metadata service address range (`169.254.169.254` and equivalent link-local ranges) at the host firewall, security group, or network policy layer — a control that neutralizes this specific exploitation path regardless of the underlying application vulnerability and is good practice for any internet- or internally-exposed service that accepts attacker-influenceable URLs. [7] Cloud providers' newer metadata-service protections, such as AWS's Instance Metadata Service Version 2 (IMDSv2) with mandatory session tokens, should be enforced on any host running MLflow or similar MLOps tooling, since IMDSv2 substantially raises the bar for SSRF-based metadata theft even against unpatched applications.

MLflow Tracking Servers should not be directly reachable from the public internet; access should be gated behind a VPN, identity-aware proxy, or equivalent authenticated network control, consistent with how organizations already treat other internal developer tooling with broad service-account access. Organizations should also restrict webhook creation permissions within MLflow to trusted administrative roles where the deployed version supports it, reducing the population of users who could register a malicious webhook even after patching closes the unauthenticated attack surface.

Strategic Considerations

CVE-2026-64849 is the latest in a pattern of 2026 disclosures — alongside SSRF and path-traversal flaws in other AI development and orchestration platforms — in which internet-exposed AI/MLOps tooling running with attached cloud identity becomes a direct pivot into an organization's cloud control plane. Security teams should treat MLOps infrastructure, including experiment-tracking servers, model registries, and pipeline orchestrators, with the same identity and network scrutiny applied to production application servers, rather than as internal developer conveniences exempt from perimeter controls. Any service architecture that accepts a URL from a user or API caller and then makes a server-side HTTP request to it — webhooks, callback URLs, "test connection" features, and similar patterns — should be assumed to require SSRF defenses that validate the destination at connection time, not merely at initial input validation, since redirect- and DNS-rebinding-based bypasses of hostname-only checks have now been demonstrated repeatedly across unrelated products.

Organizations running MLOps platforms should also revisit the IAM roles and service accounts attached to those hosts under a least-privilege standard. An SSRF vulnerability's real-world impact is bounded by what the underlying host's cloud identity can do; a Tracking Server whose attached role can only write to a single, scoped model-artifact bucket presents a materially smaller blast radius than one running with broad account-level permissions, even when both are equally vulnerable to the SSRF itself.

CSA Resource Alignment

CVE-2026-64849 fits a broader pattern CSA has tracked across AI and MLOps tooling: internet-exposed services that accept a URL or hostname as input and then make a server-side request to it are a recurring vector for cloud-credential theft whenever validation checks the destination only once and trusts it thereafter. Addressing this class of risk starts with inventorying every service that accepts untrusted URL or hostname input, then favoring resolved-IP-range allowlists and connection-time validation over domain-based checks, routing outbound traffic through egress controls that block metadata endpoints, and enforcing least-privilege, workload-specific cloud identities rather than broad default service accounts. These practices map directly onto the MLflow case and should guide both the immediate response and any broader review of MLOps infrastructure.

CSA's research note "Langflow Path Traversal: Unauthenticated RCE Actively Exploited" documented a structurally similar problem in a different AI development platform: an internet-exposed, credential-rich AI tool with a permissive default posture became an actively exploited pivot into an organization's broader environment within days of disclosure. [8] Read together, these cases support treating internet-exposed MLOps and AI-development platforms as a distinct asset class requiring dedicated inventory, patch-SLA, and network-exposure governance, rather than folding them into generic developer-tooling risk categories.

The AI Controls Matrix (AICM), CSA's control framework for AI systems, provides the governance baseline for closing the gaps this CVE exposes. [9] Its threat-and-vulnerability-management and application-security-related control domains cover the SSRF validation weakness itself, while its identity-and-access-management domain addresses the least-privilege and credential-scoping questions raised by MLflow's typical deployment with attached cloud IAM roles. Organizations conducting AICM-aligned assessments of their MLOps environments should treat webhook and callback-URL features across their AI toolchain as a specific control-evaluation point, given how frequently this bug class recurs.

References

[1] NIST National Vulnerability Database. "CVE-2026-64849 Detail." NVD, August 2026.

[2] MLflow. "DNS Rebinding SSRF Vulnerability in Webhook Delivery — Issue #24179." GitHub, August 2026.

[3] MLflow. "Fix: Connection-Time SSRF Protection for Webhook Delivery — Pull Request #24258." GitHub, August 2026.

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

[5] Rescana. "Active Exploitation of MLflow SSRF Vulnerability (CVE-2026-64849) Enables Cloud Credential Theft and Account Compromise." Rescana, August 2026.

[6] CISA. "Known Exploited Vulnerabilities Catalog." U.S. Cybersecurity and Infrastructure Security Agency, August 19, 2026.

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

[8] Cloud Security Alliance. "Langflow Path Traversal: Unauthenticated RCE Actively Exploited." CSA, 2026.

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

