INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-04-22 18:51 UTC

LMDeploy SSRF Vulnerability in Vision-Language Module Allows Internal Network Access

CVE VULNERABILITY | **HIGH** | CVSS 8.6 | **CISA KEV**

SCC Item ID	SCC-CVE-2026-0070
Type	CVE Vulnerability
CVE ID	CVE-2026-33626
Severity	HIGH
CVSS Base Score	8.6
EPSS Score	0.0003 (9th percentile)
KEV Status	Yes — CISA Known Exploited Vulnerability
Affected Products	internlm/lmdeploy < 0.12.3
Published	2026-04-22T00:00:00Z
Discovery Source	Vulncheck Kev

Executive Summary

A confirmed-exploited vulnerability in LMDeploy, an open-source AI model serving framework, allows attackers to manipulate the server into accessing internal network resources, including cloud infrastructure metadata services. Organizations deploying LMDeploy versions prior to 0.12.3 in cloud or hybrid environments face risk of IAM credential exfiltration and lateral movement into internal systems. This vulnerability is listed on the CISA Known Exploited Vulnerabilities catalog, indicating active exploitation in the wild.

Technical Analysis

CVE-2026-33626 is a Server-Side Request Forgery (SSRF) vulnerability (CWE-918) in LMDeploy's vision-language module, affecting all versions prior to 0.12.3. The vulnerable function, load_image() in lmdeploy/vl/utils.py, fetches user-supplied image URLs without validating or blocking requests to private or internal IP ranges. An attacker submitting a crafted image URL can cause the server to issue outbound HTTP requests to cloud metadata services such as AWS IMDSv1 (169.254.169.254), internal hosts, or other non-public resources. Successful exploitation enables retrieval of IAM credentials, internal service enumeration, and potential lateral movement. MITRE ATT&CK mappings: T1552.005 (Cloud Instance Metadata API) and T1190 (Exploit Public-Facing Application). CVSS base score: 8.6. EPSS: 0.0031% (9th percentile). The vulnerability is confirmed actively exploited per VulnCheck KEV and listed on CISA KEV. The fix is available in

version 0.12.3.

Action Checklist

1. Step 1: Containment, Immediately identify all LMDeploy deployments running versions prior to 0.12.3. Restrict inbound image URL submissions at the application or WAF layer. Block outbound server requests to 169.254.169.254 and RFC-1918 address ranges (10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16) via host-based firewall or network egress controls on any host running LMDeploy.
2. Step 2: Detection, Review outbound HTTP/HTTPS request logs from LMDeploy host processes for requests to 169.254.169.254 or other link-local and private IP ranges. Check cloud provider access logs (AWS CloudTrail, Azure Activity Log, GCP Cloud Audit Logs) for unexpected metadata API calls or credential usage from the LMDeploy service account. Look for anomalous IAM API calls or new access keys created near the time of any suspicious metadata request.
3. Step 3: Eradication, Upgrade LMDeploy to version 0.12.3 or later per the official internlm/lmdeploy repository (https://github.com/InternLM/lmdeploy/releases/tag/v0.12.3). Confirm the patched load_image() function is active post-upgrade. Rotate any cloud IAM credentials, instance profile keys, or service account tokens accessible from affected hosts.
4. Step 4: Recovery, After patching and credential rotation, validate that outbound requests to metadata endpoints no longer succeed from LMDeploy processes. Monitor cloud provider access logs for 7 to 14 days post-remediation for any continued anomalous credential use that may indicate prior credential exfiltration. Confirm application functionality is intact following the version upgrade.
5. Step 5: Post-Incident, Review egress filtering controls for all AI/ML model-serving infrastructure; metadata endpoint access should be restricted by default, not only after an incident. Evaluate whether the SSRF risk class (CWE-918) is addressed in your secure development and vendor evaluation standards. Add SSRF-specific detection rules to your SIEM for ongoing coverage of this attack pattern across other services.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate immediately to CISO and legal/privacy counsel if CloudTrail, Azure Activity Log, or GCP Cloud Audit Logs show any successful GetCredentials, AssumeRole, or equivalent metadata API response from the LMDeploy service account, as this confirms credential exfiltration and triggers cloud breach notification assessment, potential regulatory reporting obligations (GDPR 72-hour window, state breach laws), and requires forensic preservation of all affected cloud IAM activity logs before any remediation destroys evidence.

Recovery Notes	After patching to LMDeploy 0.12.3 and rotating all cloud IAM credentials accessible from affected hosts, conduct a controlled SSRF validation test against a non-production replica to confirm `load_image()` now rejects requests to `169.254.169.254` and RFC-1918 ranges before returning any LMDeploy host to production traffic. Monitor CloudTrail, Azure Activity Log, or GCP Cloud Audit Logs for the rotated service account and instance profile for a minimum of 14 days post-remediation, specifically for `CreateAccessKey`, `AssumeRole`, `GetSessionToken`, or equivalent events that would indicate pre-rotation credentials were exfiltrated and are being used by a threat actor from external infrastructure. Confirm LMDeploy vision-language inference endpoints return expected model outputs on benign image inputs before declaring recovery complete, as the patch modifies `load_image()` behavior and could affect URL-sourced image ingestion workflows.
Forensic Artifacts	LMDeploy application logs containing HTTP requests where the image_url parameter or vision-language API input field targets http://169.254.169.254/latest/meta-data/iam/security-credentials/ — the direct exploit artifact for CVE-2026-33626's load_image() SSRF vector; location varies by deployment but check systemd journal (journalctl -u lmdeploy) and any --log-level debug output files | AWS CloudTrail, Azure Activity Log, or GCP Cloud Audit Log records showing GetCredentials, GetSessionToken, AssumeRole, or equivalent metadata API calls originating from the LMDeploy host's instance identity or service account during the exploitation window — successful responses confirm credential exfiltration | Kernel-level network connection audit records from auditd (syscall=connect filter) or tcpdump/pcap captures showing outbound TCP connections from the LMDeploy worker process PID to 169.254.169.254:80 or RFC-1918 destinations, which are anomalous for an AI inference process and directly evidence SSRF exploitation | Pre-patch load_image() Python source preserved from the installed lmdeploy package (extracted via inspect.getsource) showing the absence of URL scheme validation or SSRF protection — serves as the technical evidence of the vulnerable code path and establishes the exploitability of the specific version deployed | IAM access key creation or service account key issuance records (AWS: iam:CreateAccessKey events in CloudTrail; GCP: iam.serviceAccounts.keys.create in Cloud Audit Logs) timestamped within minutes of any confirmed metadata endpoint access — this correlation is the primary indicator that SSRF exploitation resulted in credential material being used to establish persistence

Per-Action IR Details

Step 1: Containment — Immediately identify all LMDeploy deployments running versions prior to 0.12.3. Restrict inbound image URL submissions at the application or WAF layer. Block outbound server requests to 169.254.169.254 and RFC-1918 address ranges (10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16) via host-based firewall or network egress controls on any host running LMDeploy.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment Strategy

Controls: NIST IR-4 (Incident Handling), NIST SC-7 (Boundary Protection), NIST SI-4 (System Monitoring), CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 4.5 (Implement and Manage a Firewall on End-User Devices), CIS 12.2 (Establish and Maintain a Secure Network Architecture)

Compensating: Run `pip show lmdeploy | grep Version` on all candidate hosts or query your asset inventory with `grep -r 'lmdeploy' /opt /usr /home --include='*.txt' -l` to surface installs. Block 169.254.169.254 immediately via iptables: `iptables -A OUTPUT -d 169.254.254.0/24 -j DROP && iptables -A OUTPUT -d 10.0.0.0/8 -j DROP && iptables -A OUTPUT -d 172.16.0.0/12 -j DROP && iptables -A OUTPUT -d 192.168.0.0/16 -j DROP` then persist with `iptables-save`. Use nftables on systemd hosts as an alternative. For WAF-less environments, deploy an nginx reverse proxy in front of LMDeploy with a deny rule on any image URL body parameter targeting RFC-1918 or link-local destinations using a Lua or ngx_http_access_module regex match.

Evidence: Before blocking egress, capture a full snapshot of active outbound connections from LMDeploy worker processes using `ss -tnp | grep lmdeploy` and `netstat -anp | grep `. Dump current iptables state (`iptables -L -n -v`) and any existing cloud security group egress rules for the affected instance. Preserve the LMDeploy process tree (`ps auxf`) to identify the service account and parent process. Pull the LMDeploy application config file (typically `~/.lmdeploy/config.json` or the deployment YAML) to confirm which vision-language endpoints are exposed and whether image URL submission is enabled. These establish the pre-containment attack surface baseline.

Step 2: Detection — Review outbound HTTP/HTTPS request logs from LMDeploy host processes for requests to 169.254.169.254 or other link-local and private IP ranges. Check cloud provider access logs (AWS CloudTrail, Azure Activity Log, GCP Cloud Audit Logs) for unexpected metadata API calls or credential usage from the LMDeploy service account. Look for anomalous IAM API calls or new access keys created near the time of any suspicious metadata request.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis

Controls: NIST AU-2 (Event Logging), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST AU-12 (Audit Record Generation), NIST SI-4 (System Monitoring), NIST IR-5 (Incident Monitoring), CIS 8.2 (Collect Audit Logs), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: On the LMDeploy host, extract outbound HTTP requests to metadata IPs from application logs using: `grep -E '169\.254\.169\.254|10\.|172\.(1[6-9]|2[0-9]|3[01])\.|192\.168\.' /var/log/lmdeploy/*.log /var/log/syslog`. Enable kernel-level network auditing with auditd: `auditctl -a always,exit -F arch=b64 -S connect -k ssrf_connect` then review with `ausearch -k ssrf_connect`. For AWS, pull CloudTrail with the AWS CLI: `aws cloudtrail lookup-events --lookup-attributes AttributeKey=EventName,AttributeValue=GetCredentials --start-time ` and `aws iam list-access-keys --user-name `. For GCP: `gcloud logging read 'protoPayload.serviceName="iamcredentials.googleapis.com"' --limit 200`. Use Wireshark or tcpdump on the LMDeploy host interface during a test request: `tcpdump -i any -nn host 169.254.169.254 -w ssrf_capture.pcap`.

Evidence: Collect LMDeploy application-level access logs (path varies by deployment; check `--log-level` startup args or systemd journal: `journalctl -u lmdeploy --since '48 hours ago'`). Extract the specific HTTP request records where the `image_url` parameter or equivalent vision-language API field contains `http://169.254.169.254/latest/meta-data/iam/security-credentials/` or RFC-1918 destinations — this is the direct exploit artifact for CVE-2026-33626's `load_image()` SSRF vector. Pull AWS CloudTrail `GetCredentials`, `AssumeRole`, and `CreateAccessKey` events tied to the instance profile or service account used by LMDeploy. Capture GCP Service Account Token requests via Cloud Audit Logs filtering on `iam.serviceAccounts.actAs`. Preserve the raw HTTP response bodies if logged, as successful SSRF against the metadata endpoint returns JSON-formatted temporary credentials — their presence confirms exploitation.

Step 3: Eradication — Upgrade LMDeploy to version 0.12.3 or later per the official internlm/lmdeploy repository. Confirm the patched load_image() function is active post-upgrade. Rotate any cloud IAM credentials, instance profile keys, or service account tokens accessible from affected hosts.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST SI-2 (Flaw Remediation), NIST IA-5 (Authenticator Management), NIST CM-6 (Configuration Settings), CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: Upgrade via pip in the LMDeploy virtual environment: `pip install --upgrade lmdeploy==0.12.3` then verify with `python -c 'import lmdeploy; print(lmdeploy.__version__)'`. To confirm the patched `load_image()` function is active, inspect the installed source: `python -c 'import inspect, lmdeploy; print(inspect.getsourcefile(lmdeploy.vl.utils.load_image))'` and review the function for URL scheme validation or allowlist enforcement that was absent in prior versions. For credential rotation on AWS: `aws iam delete-access-key --access-key-id ` and `aws iam create-access-key`, then update all secrets referencing the old key. For GCP, rotate the service account key via `gcloud iam service-accounts keys delete --iam-account=` and issue a new key. Verify no hardcoded credentials exist in LMDeploy config files or environment variables post-rotation.

Evidence: Before upgrading, preserve the vulnerable `load_image()` function source for forensic record: `python -c 'import inspect, lmdeploy; print(inspect.getsource(lmdeploy.vl.utils.load_image))' > load_image_preupgrade.txt`. Document the exact installed version hash with `pip show lmdeploy` and `pip download lmdeploy== --no-deps -d ./evidence/` to retain the vulnerable artifact. Pull the full list of IAM credentials active during the exploitation window from CloudTrail before rotation — any `CreateAccessKey` or `GetSessionToken` events from the LMDeploy host after a metadata request indicate already-exfiltrated credentials that require revocation, not just rotation.

Step 4: Recovery — After patching and credential rotation, validate that outbound requests to metadata endpoints no longer succeed from LMDeploy processes. Monitor cloud provider access logs for 7 to 14 days post-remediation for any continued anomalous credential use that may indicate prior credential exfiltration. Confirm application functionality is intact following the version upgrade.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST IR-4 (Incident Handling), NIST SI-6 (Security and Privacy Function Verification), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST CP-10 (System Recovery and Reconstitution), CIS 8.2 (Collect Audit Logs), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Validate SSRF is closed by sending a controlled test request through the LMDeploy vision-language API with `image_url` set to `http://169.254.169.254/latest/meta-data/` and confirming the server rejects it or returns an error — do NOT do this against production without authorization; use a staging replica. From the host, confirm the block is enforced: `curl --max-time 3 http://169.254.169.254/latest/meta-data/ 2>&1` should timeout or be refused after egress rules are in place. For continued credential monitoring, set a CloudWatch Alarm (AWS free tier) on CloudTrail metric filter matching `{ $.eventName = "GetCredentials" && $.userIdentity.arn = "" }` with a threshold of 0 expected calls post-rotation. For GCP, use Cloud Monitoring with a log-based alert on the rotated service account's token activity.

Evidence: Preserve the post-patch `load_image()` source as the verified clean baseline alongside the pre-patch version captured during eradication. Log the exact timestamp and actor for each rotated credential and retain CloudTrail/Activity Log records showing the old keys were deactivated before any new usage. During the 7–14 day monitoring window, collect and archive all IAM authentication events for the LMDeploy service account to provide an audit trail demonstrating no post-incident credential abuse — this record supports both internal lessons learned (NIST 800-61r3 §4) and any regulatory notification assessment tied to cloud data access.

Step 5: Post-Incident — Review egress filtering controls for all AI/ML model-serving infrastructure; metadata endpoint access should be restricted by default, not only after an incident. Evaluate whether the SSRF risk class (CWE-918) is addressed in your secure development and vendor evaluation standards. Add SSRF-specific detection rules to your SIEM for ongoing coverage of this attack pattern across other services.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity

Controls: NIST IR-4 (Incident Handling), NIST IR-8 (Incident Response Plan), NIST SI-10 (Information Input Validation), NIST RA-3 (Risk Assessment), NIST SA-11 (Developer Testing and Evaluation), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process), CIS 4.4 (Implement and Manage a Firewall on Servers)

Compensating: Deploy a Sigma rule targeting SSRF-to-metadata patterns across all HTTP-capable services, not just LMDeploy — use the community Sigma rule `proxy_ssrf_localhost` as a base and extend the destination filter to include `169.254.169.254`, `100.100.100.200` (Alibaba Cloud metadata), and `192.0.2.0/24`. Add this to your log review checklist if you lack a SIEM. For vendor evaluation, create a one-page SSRF checklist based on CWE-918 to be completed before onboarding any new AI/ML serving framework: does the product validate URL schemes in user-supplied inputs? Does it restrict outbound connections from inference processes? For egress hardening across all ML serving hosts, codify the iptables rules from Step 1 into a hardening script and run it via Ansible or a cron-deployed shell script on a schedule. Document the LMDeploy CVE-2026-33626 incident in your lessons-learned register and use it as a tabletop scenario for testing SSRF detection gaps in other internal services.

Evidence: Compile the full incident timeline from logs collected in Steps 1–4: first SSRF attempt timestamp from LMDeploy application logs, first metadata API hit from CloudTrail/Activity Logs, credential rotation timestamps, and patch deployment confirmation. This timeline is the primary artifact for the post-incident report per NIST 800-61r3 §4 and supports any regulatory disclosure assessment. Retain the pre-patch `load_image()` source, the captured network pcap (if collected), and the IAM audit records as a complete evidence package for the incident file.

Detection Guidance

Query outbound network logs and proxy logs for HTTP GET requests originating from LMDeploy process accounts targeting 169.254.169.254, 169.254.170.2 (AWS ECS metadata), or any RFC-1918 address space. In AWS environments, search CloudTrail for GetCallerIdentity, AssumeRole, or ListAccessKeys events sourced from the LMDeploy instance's IAM role outside of expected application activity. In Kubernetes environments, check for unexpected requests to the Kubernetes API server from the pod running LMDeploy. Behavioral indicator: a crafted image URL submitted as input to the vision-language module that resolves to a non-public IP. No public IOC hashes or C2 domains have been confirmed for this vulnerability at this time.

Indicators of Compromise

Type	Value	Context	Confidence
IP	`169.254.169.254`	AWS IMDSv1 metadata endpoint targeted by SSRF exploitation of CVE-2026-33626; outbound requests to this IP from LMDeploy hosts indicate exploitation attempt or success	HIGH

Framework Mappings

MITRE-ATTACK

- **T1552.005** — Cloud Instance Metadata API
- **T1190** — Exploit Public-Facing Application

NIST-800-53R5

- **CA-8** — Penetration Testing
- **RA-5** — Vulnerability Monitoring and Scanning
- **SC-7** — Boundary Protection
- **SI-2** — Flaw Remediation
- **SI-7** — Software, Firmware, and Information Integrity
- **SI-10** — Information Input Validation

OWASP-TOP10-2021

- **A10:2021** — Server-Side Request Forgery (SSRF)

CIS-V8

- **13.4** — Perform Traffic Filtering Between Network Segments

- **7.3** — Perform Automated Operating System Patch Management
- **7.4** — Perform Automated Application Patch Management

ISO-27001-2022

- **A.8.8** — Management of technical vulnerabilities
- **A.5.23** — Information security for use of cloud services

NIST-CSF-2

- **DE.AE-08** — Incidents are declared when adverse events meet the defined incident criteria

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1552.005	Cloud Instance Metadata API	Credential-Access
T1190	Exploit Public-Facing Application	Initial-Access

Sources

Source	URL	Tier
vulncheck_kev	https://nvd.nist.gov/vuln/detail/CVE-2026-33626	T1
CVE-2026-33626 | Tenable®	https://www.tenable.com/cve/CVE-2026-33626	T3
CVE-2026-33626 | Mondoo Vulnerability Intelligence	https://mondoo.com/vulnerability-intelligence/vulnerability/CVE-202...	T3
CVE-2026-33626: LMDeploy VL Image Loading SSRF | Miggo	https://www.miggo.io/vulnerability-database/cve/CVE-2026-33626	T3
CVE-2026-33626 Security Vulnerability Analysis & Exploit Details	https://cve.akaoma.com/cve-2026-33626	T3
CISA KEV	https://www.cisa.gov/known-exploited-vulnerabilities-catalog	T1

Generated 2026-04-22 18:51 UTC by TJS Security Command Center