INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-04-13 16:29 UTC

GHSA-fvcv-3m26-pcqx: Axios has Unrestricted Cloud Metadata Exfiltration via Header Injection Chain

CVE VULNERABILITY | **CRITICAL** | CVSS 9.1

SCC Item ID	SCC-CVE-2026-0036
Type	CVE Vulnerability
CVE ID	CVE-2026-40175
Severity	CRITICAL
CVSS Base Score	9.1
EPSS Score	0.0024 (47th percentile)
Affected Products	axios (npm), specific affected versions not confirmed from available sources
Published	2026-04-10T19:47:16Z
Discovery Source	Osv

Executive Summary

A critical vulnerability in Axios, one of the most widely used HTTP client libraries in JavaScript and Node.js applications, allows attackers to hijack outbound requests and steal cloud credentials, including AWS IAM roles and access tokens, from the underlying infrastructure. Any organization running Axios in a cloud-hosted application may be exposed to unauthorized access to cloud accounts, data stores, and downstream services. Exploitation does not require authentication and the attack path is well understood, making this a high-priority patching target. While specific affected versions are pending NVD publication, organizations should audit all Axios dependencies as a precaution.

Technical Analysis

CVE-2026-40175 (GHSA-fvcv-3m26-pcqx) is a critical-rated vulnerability in the Axios npm library combining HTTP header injection (CWE-93) and Server-Side Request Forgery (CWE-918). An attacker who can influence request headers passed through Axios can inject a crafted header that redirects the HTTP request to the cloud instance metadata service (IMDS), specifically the AWS IMDSv1 endpoint at 169.254.169.254 (GCP metadata.google.internal, Azure 169.254.169.254). Successful exploitation exposes IAM role credentials, temporary access tokens, and environment configuration without authentication. CVSS base score is 9.1 (Critical). EPSS score is 0.00239 (46.96th percentile), indicating low current exploitation activity but a

well-understood attack class. MITRE ATT&CK coverage: T1552.005 (Cloud Instance Metadata API), T1599 (Network Boundary Bridging), T1071.001 (Web Protocols). Affected version ranges and fixed versions are available from the OSV database entry (GHSA-fvcv-3m26-pcqx). The critical rating is corroborated across multiple vulnerability intelligence sources. IMDSv2 enforcement on AWS mitigates the credential theft vector but does not eliminate the header injection risk. CWE-93 and CWE-918 are both present; remediation requires patching Axios to the fixed release, not solely relying on infrastructure controls.

Action Checklist

1. Step 1: Containment, Identify all services and pipelines that include Axios as a direct or transitive npm dependency. Run 'npm list axios' or 'npm ls axios --all' across repositories and deployed environments. If affected versions are confirmed, temporarily restrict outbound HTTP from those services to internal metadata IP ranges (169.254.169.254, 169.254.170.2) at the network or security group level until patching is complete.
2. Step 2: Detection, Query application logs and WAF logs for outbound requests to 169.254.169.254 or requests containing IMDS API paths (/latest/meta-data/, /latest/dynamic/instance-identity/). In AWS environments, enable VPC Flow Logs and search for traffic to 169.254.169.254 from application subnets. In CloudTrail, search for AssumeRole or GetSessionToken events that do not match expected service principals (compare against your baseline of approved service principals for this application role). Flag anomalous header values in API gateway or reverse proxy access logs.
3. Step 3: Eradication, Upgrade Axios to the patched release version as specified in the GHSA-fvcv-3m26-pcqx advisory on OSV (https://osv.dev/vulnerability/GHSA-fvcv-3m26-pcqx). Confirm fixed version in package-lock.json or yarn.lock after upgrade. If affected version ranges are confirmed via NVD before patching, prioritize services with internet-facing request handling. Enforce IMDSv2 (session-oriented tokens) on all EC2 instances as a defense-in-depth control; this blocks credential theft via IMDSv1 SSRF even if header injection is not fully eliminated by older library versions.
4. Step 4: Recovery, After upgrading, re-run dependency audits ('npm audit') to confirm no residual Axios versions remain via transitive dependencies. Verify outbound request behavior in staging before promoting to production. Monitor CloudTrail and IMDS access logs for 48-72 hours post-patch for any signs of prior successful exploitation. Rotate IAM credentials and temporary tokens for roles associated with affected services as a precaution if IMDS access was detected.
5. Step 5: Post-Incident, This vulnerability exposes two control gaps: (1) insufficient input validation on HTTP headers passed to client libraries, and (2) reliance on IMDSv1 rather than IMDSv2. Review SDLC controls to enforce dependency pinning and automated CVE scanning in CI/CD pipelines (NIST SP 800-218 SSDF practice PW.4). Evaluate adoption of IMDSv2 enforcement as a baseline cloud configuration standard. Map to NIST CSF 2.0 Govern (GV.SC) and Protect (PR.DS) functions for supply chain and data protection control review.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE

Escalation Criteria	Escalate to CISO and legal/compliance immediately if CloudTrail confirms any AssumeRole, GetSessionToken, or data-plane API call (S3:GetObject, RDS:Connect, SecretsManager:GetSecretValue) originating from an application subnet IP during the exposure window, as this indicates successful IAM credential theft and may trigger breach notification obligations under applicable data protection regulations.
Recovery Notes	After patching Axios and enforcing IMDSv2, monitor CloudTrail for a minimum of 72 hours for any IAM API calls from application subnet source IPs or using role ARNs associated with affected services — stolen IMDSv1 tokens issued before the patch may still be valid until their TTL expires (typically 6 hours for STS temporary credentials, but up to 12 hours for some role configurations). If IMDS access was confirmed, treat all IAM temporary credentials issued to affected roles during the exposure window as compromised and force rotation by detaching and reattaching instance profiles. Verify that 'npm audit' returns zero CRITICAL findings and that all deployed package-lock.json files pin Axios at the patched version before restoring unrestricted outbound network access.
Forensic Artifacts	VPC Flow Logs — ACCEPT records for TCP flows from application EC2 instance ENIs to 169.254.169.254:80; presence of ACCEPT (not REJECT) confirms successful IMDS queries, not merely blocked attempts, and establishes whether IMDSv1 was reachable at time of exploitation | AWS CloudTrail event history — AssumeRole, GetSessionToken, and GetCallerIdentity events where sourceIPAddress matches application subnet CIDR ranges rather than AWS service endpoints; this is the primary indicator that Axios header injection successfully redirected a request to IMDS and the returned credentials were used | Node.js application logs or stdout — outbound request URLs logged by Axios interceptors or Node.js HTTP module containing '169.254.169.254' or IMDS path strings ('/latest/meta-data/iam/security-credentials/'); these capture the header injection payload as it was executed by the vulnerable Axios version | AWS API Gateway or ALB access logs — inbound HTTP request records containing crafted header values (e.g., 'Host: 169.254.169.254', 'X-Forwarded-Host: 169.254.169.254', or URL-encoded IMDS paths in header fields) that represent the attacker-supplied injection payload delivered to the vulnerable Axios-consuming service | package-lock.json and yarn.lock snapshots from deployed artifacts — document the exact Axios version (and the dependency chain through which it was introduced) present on affected hosts at time of exploitation; critical for scope assessment and for identifying all other services sharing the same transitive dependency path

Per-Action IR Details

Step 1: Containment — Identify all services and pipelines that include Axios as a direct or transitive npm dependency. Run 'npm list axios' or 'npm ls axios --all' across repositories and deployed environments. If affected versions are confirmed, temporarily restrict outbound HTTP from those services to internal metadata IP ranges (169.254.169.254, 169.254.170.2) at the network or security group level until patching is complete.

NIST Phase: Containment

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

Controls: NIST IR-4 (Incident Handling), NIST CM-7 (Least Functionality), NIST SC-7 (Boundary Protection), CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 2.1 (Establish and Maintain a Software Inventory)

Compensating: Run 'npm ls axios --all 2>/dev/null | grep axios' recursively across all Node.js project roots using a simple bash loop: 'find / -name package.json -not -path "*/node_modules/*" -exec dirname {} \; | xargs -I{} bash -c "cd {} && npm ls axios --all 2>/dev/null | grep axios && echo {}"'. Block 169.254.169.254 and 169.254.170.2 using AWS Security Group outbound rules (remove or restrict egress rules for port 80/TCP to these CIDRs) or iptables: 'iptables -A OUTPUT -d 169.254.169.254 -j DROP && iptables -A OUTPUT -d 169.254.170.2 -j DROP'. For container workloads, apply the same rule via Docker network policy or ECS task security groups.

Evidence: Before restricting egress, capture the current AWS Security Group outbound rules and VPC Flow Logs for the past 72 hours targeting 169.254.169.254 (TCP port 80) from application subnet CIDRs. Export 'npm ls axios --all' output from each affected service as a dependency tree artifact. Pull current EC2 instance metadata access logs from IMDSv1 endpoint if accessible via 'curl http://169.254.169.254/latest/meta-data/' from each host — a successful response confirms IMDSv1 is still active and exploitable. Preserve these artifacts before any network changes remove evidence of prior access attempts.

Step 2: Detection — Query application logs and WAF logs for outbound requests to 169.254.169.254 or requests containing IMDS API paths (/latest/meta-data/, /latest/dynamic/instance-identity/). In AWS environments, enable VPC Flow Logs and search for traffic to 169.254.169.254 from application subnets. In CloudTrail, search for AssumeRole or GetSessionToken events that do not match expected service principals. Flag anomalous header values in API gateway or reverse proxy access logs.

NIST Phase: Detection Analysis

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

Controls: NIST IR-5 (Incident Monitoring), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST AU-12 (Audit Record Generation), NIST SI-4 (System Monitoring), CIS 8.2 (Collect Audit Logs)

Compensating: Query CloudTrail logs directly from S3 using AWS CLI without a SIEM: 'aws cloudtrail lookup-events --lookup-attributes AttributeKey=EventName,AttributeValue=AssumeRole --start-time $(date -d "7 days ago" +%s) --query "Events[?Username!='expected-service-account']" --output json'. For VPC Flow Logs stored in S3, use AWS CLI with grep: 'aws s3 cp s3://your-flow-log-bucket/ . --recursive --include "*.gz" && zcat *.gz | grep "169.254.169.254"'. For application log analysis without a SIEM, use 'grep -rE "169\.254\.169\.254|/latest/meta-data|/latest/dynamic/instance-identity"' against Nginx/Apache access logs. Use the open-source Sigma rule for SSRF against IMDS (search SigmaHQ GitHub for 'aws_imds_ssrf') converted to a jq query against CloudTrail JSON exports.

Evidence: Capture and preserve: (1) CloudTrail event history JSON for AssumeRole, GetSessionToken, and GetCallerIdentity events for the past 30 days — specifically looking for source IPs matching application subnet ranges rather than expected IAM service endpoints; (2) VPC Flow Logs showing TCP connections to 169.254.169.254:80 from application EC2 instance ENIs — the presence of ACCEPT records confirms successful IMDS queries, not just attempts; (3) Application-layer logs from Node.js services running Axios showing outbound request URLs containing the string '169.254.169.254' or header values like 'X-Forwarded-Host', 'X-Forwarded-For', or 'Host' set to the IMDS address, which is the specific injection artifact this vulnerability produces; (4) AWS API Gateway or ALB access logs for requests containing crafted header payloads targeting Axios header injection.

Step 3: Eradication — Upgrade Axios to the patched release version as specified in the GHSA-fvcv-3m26-pcqx advisory on OSV (https://osv.dev/vulnerability/GHSA-fvcv-3m26-pcqx). Confirm fixed version in package-lock.json or yarn.lock after upgrade. If affected version ranges are confirmed via NVD before patching, prioritize services with internet-facing request handling. Enforce IMDSv2 (session-oriented tokens) on all EC2 instances as a defense-in-depth control — this blocks credential theft via IMDSv1 SSRF even if header injection succeeds.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST SI-2 (Flaw Remediation), NIST CM-7 (Least Functionality), NIST SA-11 (Developer Testing and Evaluation), 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: Enforce IMDSv2-only on each EC2 instance without automation tooling using AWS CLI: 'aws ec2 modify-instance-metadata-options --instance-id i-XXXXXXXX --http-tokens required --http-put-response-hop-limit 1 --region us-east-1'. Verify IMDSv1 is blocked by confirming 'curl http://169.254.169.254/latest/meta-data/' returns a 401 from within the instance. For package upgrade verification, run 'npm audit --json | jq ".vulnerabilities | to_entries[] | select(.value.via[].source == \"GHSA-fvcv-3m26-pcqx\")"' to confirm the advisory no longer appears post-patch. For transitive dependency pinning, add an explicit 'axios' entry in the 'overrides' (npm) or 'resolutions' (yarn) field of package.json to force the patched version across the entire dependency tree.

Evidence: Before patching, snapshot the exact contents of package-lock.json and yarn.lock from all affected services to document the vulnerable Axios version chain (direct and transitive). Export 'npm audit --json' output pre-patch as a baseline artifact. Capture the EC2 instance metadata options state pre-remediation using: 'aws ec2 describe-instances --query "Reservations[].Instances[].[InstanceId,MetadataOptions]" --output json' — this documents which instances were running IMDSv1 at the time of the incident. Preserve these artifacts for post-incident review and potential breach notification evidence.

Step 4: Recovery — After upgrading, re-run dependency audits ('npm audit') to confirm no residual Axios versions remain via transitive dependencies. Verify outbound request behavior in staging before promoting to production. Monitor CloudTrail and IMDS access logs for 48-72 hours post-patch for any signs of prior successful exploitation. Rotate IAM credentials and temporary tokens for roles associated with affected services as a precaution if IMDS access was detected.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST IR-4 (Incident Handling), NIST CP-10 (System Recovery and Reconstitution), NIST IA-5 (Authenticator Management), NIST AU-6 (Audit Record Review, Analysis, and Reporting), CIS 5.2 (Use Unique Passwords), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Rotate IAM role credentials associated with affected services using: 'aws iam create-access-key --user-name service-account && aws iam delete-access-key --access-key-id OLDKEYID --user-name service-account'. For EC2 instance roles (no static keys), force token invalidation by detaching and reattaching the instance profile: 'aws ec2 disassociate-iam-instance-profile --association-id iip-assoc-XXXXXXXX && aws ec2 associate-iam-instance-profile --instance-id i-XXXXXXXX --iam-instance-profile Name=OriginalRoleName'. Monitor post-patch CloudTrail continuously for 72 hours using a scheduled AWS CLI cron job: 'aws cloudtrail lookup-events --lookup-attributes AttributeKey=EventName,AttributeValue=AssumeRole --start-time $(date -d "1 hour ago" +%s) --output json | jq ".Events[] | select(.Username != \"expected-principal\")"' run every 15 minutes and alerting on any output.

Evidence: Post-upgrade, run 'npm audit --json' and 'npm ls axios --all' on deployed artifacts and capture output as remediation evidence. Collect CloudTrail GetCallerIdentity and AssumeRole events from the 48-72 hour post-patch monitoring window to establish a clean-state baseline and identify any anomalous API calls using stolen tokens that may have been harvested prior to patching. If IMDS access was confirmed during detection, preserve the full CloudTrail event for any IAM actions taken by the compromised role — these may constitute breach evidence if sensitive data stores or downstream services were accessed using stolen tokens.

Step 5: Post-Incident — This vulnerability exposes two control gaps: (1) insufficient input validation on HTTP headers passed to client libraries, and (2) reliance on IMDSv1 rather than IMDSv2. Review SDLC controls to enforce dependency pinning and automated CVE scanning in CI/CD pipelines (NIST SP 800-218 SSDF practice PW.4). Evaluate adoption of IMDSv2 enforcement as a baseline cloud configuration standard. Map to NIST CSF 2.0 Govern (GV.SC) and Protect (PR.DS) functions for supply chain and data protection control review.

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-2 (Flaw Remediation), NIST SI-7 (Software, Firmware, and Information Integrity), 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 2.2 (Ensure Authorized Software is Currently Supported)

Compensating: Integrate free CVE scanning into CI/CD using 'npm audit --audit-level=critical' as a blocking gate in GitHub Actions or GitLab CI — a non-zero exit code on CRITICAL findings fails the pipeline. Add OSV-Scanner (free, Google-maintained) to the pipeline: 'osv-scanner --lockfile=package-lock.json' will flag GHSA advisories including header injection classes before deployment. For SDLC header validation, add a code review checklist item requiring all Axios request configurations that accept user-controlled input for 'headers', 'baseURL', or 'url' fields to pass through an allowlist validation function. Enforce IMDSv2 organization-wide using an AWS Config rule: 'ec2-imdsv2-check' (managed rule, free tier eligible) to detect any instance not configured with 'HttpTokens: required'.

Evidence: For the lessons-learned record, compile: (1) the full dependency tree showing how CVE-2026-40175 entered the environment (direct vs. transitive), (2) the timeline from advisory publication (GHSA-fvcv-3m26-pcqx) to detection, containment, and patch — this gap measurement drives SLA improvements per NIST IR-8; (3) a list of all EC2 instances that were running IMDSv1 at incident time (from the pre-remediation 'describe-instances' snapshot), which documents the blast radius of any confirmed exploitation; (4) CloudTrail evidence of any IAM credential use originating from application subnet IPs outside normal service behavior during the exposure window — this is the definitive indicator of whether token theft was successful and may trigger breach notification obligations.

Detection Guidance

Primary detection target: outbound HTTP requests to the AWS instance metadata IP 169.254.169.254 or equivalent GCP and Azure endpoints originating from application processes. In AWS: enable VPC Flow Logs and filter for destination 169.254.169.254 from application-tier subnets. In CloudTrail: search for credential-related API calls (sts:AssumeRole, sts:GetSessionToken, iam:ListRoles) from unexpected source IPs or principals. At the application layer: inspect access logs for requests with injected CRLF sequences or unexpected Host/X-Forwarded-For header values. If a WAF is in place, review for 'header injection' or 'SSRF' rule triggers against Axios-backed endpoints. Behavioral indicator: a sudden increase in outbound requests to non-business IP ranges from Node.js service processes. EPSS percentile is 46.96; active exploitation is not yet widely observed, but the attack pattern matches documented SSRF-to-IMDS techniques with no novel steps required.

Indicators of Compromise

Type	Value	Context	Confidence
IP	`169.254.169.254`	AWS IMDSv1 instance metadata endpoint — outbound requests to this IP from application processes are a primary SSRF exploitation indicator for this vulnerability	HIGH
URL	`http://169.254.169.254/latest/meta-data/iam/security-credentials/`	Specific IMDS path targeted to retrieve IAM role credentials via SSRF exploitation	HIGH
URL	`http://169.254.169.254/latest/dynamic/instance-identity/document`	IMDS path for instance identity and region data, commonly retrieved alongside credentials in SSRF attacks	MEDIUM

Framework Mappings

MITRE-ATTACK

- **T1599** — Network Boundary Bridging
- **T1552.005** — Cloud Instance Metadata API
- **T1071.001** — Web Protocols

OWASP-TOP10-2021

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

NIST-800-53R5

- **SC-7** — Boundary Protection
- **SI-10** — Information Input Validation

CIS-V8

- **13.4** — Perform Traffic Filtering Between Network Segments
- **6.3** — Require MFA for Externally-Exposed Applications

HIPAA-SECURITY

- **164.312(d)** — Person or Entity Authentication
- **164.308(a)(6)(ii)** — Response and Reporting

SOC2-TSC

- **CC6.1** — Logical access security software, infrastructure, and architectures
- **CC7.4** — Responds to identified security incidents

ISO-27001-2022

- **A.8.8** — Management of technical vulnerabilities
- **A.5.34** — Privacy and protection of personal information
- **A.5.23** — Information security for use of cloud services

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1599	Network Boundary Bridging	Defense-Evasion
T1552.005	Cloud Instance Metadata API	Credential-Access
T1071.001	Web Protocols	Command-And-Control

Sources

Source	URL	Tier
osv	https://osv.dev/vulnerability/GHSA-fvcv-3m26-pcqx	T3
CVE-2026-40175 | Mondoo Vulnerability Intelligence	https://mondoo.com/vulnerability-intelligence/vulnerability/CVE-202...	T3
Linux Distros Unpatched Vulnerability : CVE-2026-40175 | Tenable®	https://www.tenable.com/plugins/nessus/306025	T3

Source	URL	Tier
CVE-2026-40175: Axios Header Injection SSRF - Miggo Security	https://www.miggo.io/vulnerability-database/cve/CVE-2026-40175	T3
[CVE-2026-40175: CRITICAL] Cyber security alert	https://x.com/CveFindCom/status/2042704916842463556	T3
NVD	https://nvd.nist.gov/vuln/detail/CVE-2026-40175	T1