INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-06-10 07:23 UTC

Cloud Logging Defense Evasion: Five Attacker Techniques Targeting AWS and GCP Audit Infrastructure

THREAT CAMPAIGN | **HIGH** | CVSS 7.5

SCC Item ID	SCC-CAM-2026-0435
Type	Threat Campaign
Severity	HIGH
CVSS Base Score	7.5
Affected Products	AWS CloudTrail, Amazon S3, AWS KMS, Google Cloud Logging, Google Cloud Storage, Google Cloud KMS (CMEK)
Published	2026-06-09T22:00:21+00:00
Discovery Source	Rss:T1 Threatintel

Executive Summary

Palo Alto Unit 42 has documented five concrete techniques attackers use to disable or destroy cloud audit logging across AWS and Google Cloud Platform, preventing detection and response before any subsequent attack phase proceeds. Every organization running CloudTrail or Google Cloud Logging without log-protection controls is exposed; no patch exists because these are technique-and-misconfiguration class attacks, not software vulnerabilities. If an attacker executes any of these techniques successfully, your SIEM, SOAR, and CSPM tools lose their foundational data source, preventing detection and response to the active intrusion.

Technical Analysis

Unit 42 (published June 9, 2026) documents five defense evasion techniques targeting cloud audit infrastructure. No CVE is assigned; this is a technique class. Affected primitives: AWS CloudTrail (trail deletion, trail disabling), Amazon S3 (bucket-level logging suppression), AWS KMS (CMK deletion or disabling to corrupt encrypted CloudTrail log integrity), Google Cloud Logging (sink deletion or disabling), Google Cloud Storage log buckets (deletion), and Google Cloud KMS CMEK destruction to corrupt GCS log integrity. CWEs: CWE-732 (incorrect permission assignment for critical resource), CWE-345 (insufficient verification of data authenticity), CWE-284 (improper access control), CWE-778 (insufficient logging). MITRE ATT&CK techniques: T1562.008 (Disable Cloud Logs), T1562.001 (Disable or Modify Tools), T1078.004 (Valid Accounts: Cloud Accounts), T1552.005 (Cloud Instance Metadata API), T1490 (Inhibit System Recovery), T1485 (Data Destruction),

T1565.001 (Stored Data Manipulation), T1046 (Network Service Discovery), T1530 (Data from Cloud Storage). Unit 42 has also documented that AI-assisted reconnaissance can accelerate discovery of these misconfigurations. IAM permission abuse is the enabling vector across all five techniques. No vendor patch is available; remediation is entirely configuration and control-based.

Action Checklist

1. Step 1: Containment. Immediately audit IAM policies across AWS and GCP to identify which principals hold CloudTrail:DeleteTrail, CloudTrail:StopLogging, s3:PutBucketLogging, kms:ScheduleKeyDeletion, kms:DisableKey, logging.sinks.delete, and storage.buckets.delete permissions. Revoke any non-break-glass (emergency-access) account holding these permissions. Apply NIST AC-6 (Least Privilege); no operational account should hold log-destruction capabilities. Reference CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts).
2. Step 2: Detection. Query CloudTrail for events: StopLogging, DeleteTrail, PutBucketLogging (with LoggingEnabled=false), ScheduleKeyDeletion, DisableKey. Query Google Cloud Audit Logs for logging.sinks.delete, storage.buckets.delete, cloudkms.cryptoKeyVersions.destroy. Alert on any of these events regardless of the calling principal; legitimate operations triggering these events should be near-zero in production. Apply NIST AU-6 (Audit Record Review, Analysis, and Reporting) and CIS 8.2 (Collect Audit Logs). Cross-reference: if a CloudTrail StopLogging event is not immediately followed by a change-management ticket, treat as confirmed evasion. Monitor IAM and log configuration for unauthorized changes.
3. Step 3: Eradication. Enable CloudTrail log file validation (integrity hashing) on all trails. Configure S3 Object Lock (WORM) on CloudTrail destination buckets to prevent deletion or overwrite. Enable MFA Delete on S3 log buckets. For AWS KMS: set key deletion windows to maximum (30 days) and require manual approval; implement Service Control Policies (SCPs) to enforce maximum deletion windows. For GCP: enable organization-level log sinks to a separate security project the primary project cannot modify. Apply AWS Service Control Policies (SCPs) to deny cloudtrail:StopLogging and cloudtrail:DeleteTrail org-wide. Apply CIS 3.3 (Configure Data Access Control Lists) and CIS 3.4 (Enforce Data Retention). Restrict and harden IAM credentials capable of log modification.
4. Step 4: Recovery. Verify all CloudTrail trails are active, delivering to S3, and log file validation is enabled. Confirm S3 bucket logging is active on each trail destination bucket. Validate GCP log sinks are delivering to an immutable destination. Run a tabletop test: attempt a simulated StopLogging API call from a non-privileged identity and confirm your SIEM fires an alert within expected SLA. Apply NIST AU-9 (Protection of Audit Information) to confirm audit data integrity controls are in place. Review SIEM data gaps from the past 90 days for any unexplained logging interruptions that may indicate prior exploitation.
5. Step 5: Post-Incident. Conduct a formal review against NIST SI-4 (System Monitoring) to assess whether current detection coverage would detect all five documented techniques. Implement alerting on CloudTrail delivery failure events (not just on the evasion API calls themselves) as a secondary detection layer. Evaluate whether your CSPM tool alerts on trail disabling; most do, but verify. Establish a monthly review cadence for IAM permissions against NIST AU-2 (Event Logging) to ensure logging-critical permissions have not drifted. Document findings in your risk register under CWE-778 (Insufficient Logging) and track remediation to closure.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate to CISO and legal counsel immediately if CloudTrail gap analysis reveals a logging blackout window exceeding 15 minutes during which sensitive data stores (S3 buckets tagged with PII, PHI, or financial data classifications) were accessible, as this may trigger breach notification obligations under HIPAA, GDPR, or state privacy statutes even absent confirmed data exfiltration — the inability to rule out access is itself a reportable condition in several jurisdictions.
Recovery Notes	After containment and eradication, treat any AWS account or GCP project where a logging-destruction event fired as potentially compromised for the full period of the logging gap — enumerate all API calls made in adjacent services (EC2, S3 data plane, IAM, RDS) during the gap window using CloudTrail data events if enabled, or AWS Config snapshots if not, to reconstruct what actions were taken while audit logging was blind. Maintain heightened monitoring on re-enabled trails for a minimum of 30 days post-recovery, specifically watching for the attacker re-attempting the same five techniques from a different IAM principal — lateral movement to a secondary credential is the most common follow-on after initial evasion tooling is burned. For GCP, verify that organization-level log sinks are receiving events by running a daily automated query against the sink destination bucket for the previous day's expected log volume, alerting on any day where ingested log count drops more than 20 percent below the 30-day rolling average.
Forensic Artifacts	AWS CloudTrail S3 digest files (s3://BUCKET/AWSLogs/ACCT/CloudTrail-Digest/REGION/YYYY/MM/DD/) — each file contains SHA-256 hashes and an HMAC chain linking consecutive digest files; gaps in the sequence or INVALID results from `aws cloudtrail validate-logs` directly prove log file tampering or suppression during specific windows tied to the five Unit 42 techniques | AWS KMS key metadata for CMKs protecting CloudTrail S3 buckets — retrieve via `aws kms describe-key` and `aws kms list-key-policies`; KeyState=PendingDeletion with a DeletionDate set confirms ScheduleKeyDeletion technique execution, and the exact scheduledDeletionDate establishes the window before which log data becomes permanently unrecoverable | GCP Cloud Audit Log Admin Activity entries for the methods logging.sinks.delete, storage.buckets.delete, and cloudkms.cryptoKeyVersions.destroy — stored in _Required log bucket which cannot be disabled or deleted even by organization admins, making this the tamper-resistant ground truth for GCP-side technique execution; extract via `gcloud logging read 'logName=projects/PROJECT/logs/cloudaudit.googleapis.com%2Factivity'` | AWS Config configuration history for the AWS::CloudTrail::Trail resource type — this records every trail configuration change with before/after state and is maintained independently of CloudTrail itself, providing a secondary timeline that remains intact even when the attacker successfully executed StopLogging or DeleteTrail; retrieve via `aws configservice get-resource-config-history --resource-type AWS::CloudTrail::Trail --resource-id TRAIL_ARN` | S3 server access logs on CloudTrail destination buckets (if bucket logging was pre-enabled) — these logs record every S3 API call including PutBucketLogging requests that disabled logging, the requesting IAM ARN, source IP, and user agent string, which together fingerprint the attacker's access method (console, CLI, SDK, or assumed role) for attribution and lateral movement tracing

Per-Action IR Details

Step 1: Containment — Immediately audit IAM policies across AWS and GCP to identify which principals hold CloudTrail:DeleteTrail, CloudTrail:StopLogging, s3:PutBucketLogging, kms:ScheduleKeyDeletion,

kms:DisableKey, logging.sinks.delete, and storage.buckets.delete permissions. Revoke any non-break-glass account holding these permissions. Apply NIST AC-6 (Least Privilege) — no operational account should hold log-destruction capabilities. Reference CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts).

NIST Phase: Containment

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

Controls: NIST AC-6 — Least Privilege, NIST AC-2 — Account Management, CIS 5.4 (IG1/IG2/IG3) — Restrict Administrator Privileges to Dedicated Administrator Accounts, CIS 5.1 (IG1/IG2/IG3) — Establish and Maintain an Inventory of Accounts

Compensating: Run `aws iam get-account-authorization-details --output json` and pipe through `jq` to extract all policies containing 'cloudtrail:StopLogging', 'cloudtrail:DeleteTrail', 'kms:ScheduleKeyDeletion', or 'kms:DisableKey'. For GCP, run `gcloud asset search-all-iam-policies --query='policy:logging.sinks.delete OR policy:storage.buckets.delete' --scope=organizations/ORG_ID`. Both commands are CLI-only, no SIEM required. Maintain a break-glass account list in a text file version-controlled in a private repo so the 2-person team can diff against current IAM state on demand.

Evidence: Before revoking any permissions, export a full snapshot of current IAM policies and CloudTrail event history for the 72-hour window preceding discovery. Specifically: (1) AWS CloudTrail management event logs filtered for the seven dangerous API calls listed in the step — capture the full event JSON including sourceIPAddress, userAgent, requestParameters, and userIdentity.arn fields; (2) AWS IAM credential report (`aws iam generate-credential-report`) to identify when implicated accounts last rotated keys or assumed roles; (3) GCP Cloud Audit Logs Admin Activity stream filtered for logging.sinks.delete and storage.buckets.delete, preserving the callerIp and authenticationInfo.principalEmail fields. Do not revoke permissions before snapshotting — revocation may trigger attacker automation or destroy evidence of the access chain.

Step 2: Detection — Query CloudTrail for events: StopLogging, DeleteTrail, PutBucketLogging (with LoggingEnabled=false), ScheduleKeyDeletion, DisableKey. Query Google Cloud Audit Logs for logging.sinks.delete, storage.buckets.delete, cloudkms.cryptoKeyVersions.destroy. Alert on any of these events regardless of the calling principal — legitimate operations triggering these events should be near-zero in production. Apply NIST AU-6 (Audit Record Review, Analysis, and Reporting) and CIS 8.2 (Collect Audit Logs). Cross-reference: if a CloudTrail StopLogging event is not immediately followed by a change-management ticket, treat as confirmed evasion. Monitor for D3-LAM (Local Account Monitoring) signals on cloud IAM accounts performing these actions.

NIST Phase: Detection Analysis

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

Controls: NIST AU-6 — Audit Record Review, Analysis, And Reporting, NIST AU-2 — Event Logging, NIST AU-12 — Audit Record Generation, CIS 8.2 (IG1/IG2/IG3) — Collect Audit Logs

Compensating: Without a SIEM, schedule an AWS CLI cron job (every 15 minutes) running: `aws cloudtrail lookup-events --lookup-attributes AttributeKey=EventName,AttributeValue=StopLogging --start-time $(date -d '15 minutes ago' --iso-8601=seconds)` for each of the five target event names, writing output to a local log file monitored by a simple grep-based alerting script that emails or pages on any non-empty result. For GCP, use `gcloud logging read 'protoPayload.methodName=~"sinks.delete|buckets.delete|cryptoKeyVersions.destroy"' --freshness=15m --format=json` on the same cadence. Publish community Sigma rules for these CloudTrail events (search sigma-rules GitHub repository under 'aws_cloudtrail_disable') to convert to whatever log pipeline is available.

Evidence: The specific artifacts this detection phase should confirm before declaring an incident: (1) CloudTrail event records with EventName=StopLogging or DeleteTrail — capture full JSON blobs including the requestID and eventTime to establish exact logging gap windows; (2) S3 server access logs on the CloudTrail destination bucket showing any PutBucketLogging API calls that set LoggingEnabled to false, with the requesting IAM ARN; (3) AWS KMS key metadata for any keys with KeyState=PendingDeletion or Disabled that protect CloudTrail S3 buckets — retrieve via `aws kms describe-key --key-id KEY_ID` and record DeletionDate; (4) GCP Cloud Audit Log entries for cloudkms.cryptoKeyVersions.destroy targeting CMEK keys protecting log storage buckets, preserving the full

protoPayload including requestMetadata.callerIp. A gap in CloudTrail event timestamps (no events for >5 minutes in a previously active trail) is itself forensic evidence of a successful StopLogging execution.

Step 3: Eradication — Enable CloudTrail log file validation (integrity hashing) on all trails. Configure S3 Object Lock (WORM) on CloudTrail destination buckets to prevent deletion or overwrite. Enable MFA Delete on S3 log buckets. For AWS KMS: set key deletion windows to maximum (30 days) and require multi-party approval via SCP. For GCP: enable organization-level log sinks to a separate security project the primary project cannot modify. Apply AWS Service Control Policies (SCPs) to deny cloudtrail:StopLogging and cloudtrail:DeleteTrail org-wide. Apply CIS 3.3 (Configure Data Access Control Lists) and CIS 3.4 (Enforce Data Retention). Implement D3-UAP (User Account Permissions) and D3-CH (Credential Hardening) to restrict and harden IAM credentials capable of log modification.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST AC-3 — Access Enforcement, NIST AC-6 — Least Privilege, NIST AU-9 — Protection Of Audit Information, NIST AU-11 — Audit Record Retention, CIS 3.3 (IG1/IG2/IG3) — Configure Data Access Control Lists, CIS 3.4 (IG1/IG2/IG3) — Enforce Data Retention

Compensating: For teams without AWS Organizations (required for SCPs), replicate the deny effect using IAM permission boundaries attached to every non-root role: create a boundary policy that explicitly denies cloudtrail:StopLogging, cloudtrail:DeleteTrail, kms:ScheduleKeyDeletion, and kms:DisableKey, then attach it to all human and service roles via a short AWS CLI loop: `for role in $(aws iam list-roles --query 'Roles[].RoleName' --output text); do aws iam put-role-permissions-boundary --role-name $role --permissions-boundary arn:aws:iam::ACCT:policy/LogProtectionBoundary; done`. For GCP without organization-level access, create a dedicated logging project with a separate billing account and grant the primary project's service accounts only logging.logWriter — never logging.sinks.delete — via a custom IAM role.

Evidence: Before applying WORM locks and SCPs, preserve the following to confirm eradication is complete and no attacker persistence remains: (1) current S3 bucket policy JSON for all CloudTrail destination buckets (`aws s3api get-bucket-policy --bucket BUCKET`) — compare against the pre-incident baseline to identify any attacker-inserted policy statements that grant external principals s3:PutBucketLogging or s3:DeleteBucket; (2) AWS KMS key policy documents for all CMKs associated with CloudTrail S3 buckets — look for any newly added key policy statements granting kms:ScheduleKeyDeletion or kms:DisableKey to unexpected principals; (3) GCP IAM policy export for the logging project (`gcloud projects get-iam-policy PROJECT_ID --format=json`) to confirm no residual bindings for logging.sinks.delete or storage.buckets.delete remain on attacker-controlled service accounts.

Step 4: Recovery — Verify all CloudTrail trails are active, delivering to S3, and log file validation is enabled. Confirm S3 bucket logging is active on each trail destination bucket. Validate GCP log sinks are delivering to an immutable destination. Run a tabletop test: attempt a simulated StopLogging API call from a non-privileged identity and confirm your SIEM fires an alert within expected SLA. Apply NIST AU-9 (Protection of Audit Information) to confirm audit data integrity controls are in place. Review SIEM data gaps from the past 90 days for any unexplained logging interruptions that may indicate prior exploitation.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST AU-9 — Protection Of Audit Information, NIST AU-6 — Audit Record Review, Analysis, And Reporting, NIST AU-3 — Content Of Audit Records, CIS 7.1 (IG1/IG2/IG3) — Establish and Maintain a Vulnerability Management Process

Compensating: Verify CloudTrail health without a SIEM by running `aws cloudtrail get-trail-status --name TRAIL_NAME` for each trail and confirming IsLogging=true and LatestDeliveryTime is within the last 15 minutes. Validate log file integrity using the AWS CLI digest validation tool: `aws cloudtrail validate-logs --trail-arn TRAIL_ARN --start-time START --end-time END` — any INVALID result confirms tampering during the gap window. For GCP, run `gcloud logging sinks describe SINK_NAME` for each sink and confirm writerIdentity has write access to the destination bucket, then pull a test log entry: `gcloud logging read 'timestamp>="RECOVERY_TIMESTAMP"' --limit=5`

to confirm live delivery.

Evidence: During recovery validation, the 90-day historical gap review should specifically target: (1) CloudTrail digest files in S3 — each hourly digest file contains a SHA-256 hash chain; missing digest files (gaps in the s3://BUCKET/AWSLogs/ACCT/CloudTrail-Digest/ prefix) pinpoint exact windows when logging was suppressed; (2) S3 access logs on the CloudTrail bucket showing any DeleteObject or PutBucketLogging API calls that do not correspond to known change management activity, which would indicate prior silent exploitation; (3) AWS Config configuration history for the CloudTrail trail resource — Config records every trail configuration change with a timestamp, providing a secondary timeline independent of CloudTrail itself for cases where an attacker disabled CloudTrail to cover a Config change.

Step 5: Post-Incident — Conduct a formal review against NIST SI-4 (System Monitoring) to assess whether current detection coverage would catch all five documented techniques. Implement alerting on CloudTrail delivery failure events (not just on the evasion API calls themselves) as a secondary detection layer. Evaluate whether your CSPM tool alerts on trail disabling — most do, but verify. Establish a monthly review cadence for IAM permissions against NIST AU-2 (Event Logging) to ensure logging-critical permissions have not drifted. Document findings in your risk register under CWE-778 (Insufficient Logging) and track remediation to closure.

NIST Phase: Post Incident

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

Controls: NIST AU-2 — Event Logging, NIST AU-6 — Audit Record Review, Analysis, And Reporting, NIST AU-13 — Monitoring For Information Disclosure, CIS 7.2 (IG1/IG2/IG3) — Establish and Maintain a Remediation Process, CIS 7.1 (IG1/IG2/IG3) — Establish and Maintain a Vulnerability Management Process

Compensating: For teams without a CSPM tool, replicate the five-technique detection coverage check using AWS Config managed rules: enable 'cloud-trail-enabled', 'cloud-trail-log-file-validation-enabled', and 'cmk-backing-key-rotation-enabled' — all are free within the AWS Config free tier up to 10 rules. For GCP, use the free Security Command Center Standard tier which includes the 'Audit logging disabled' and 'KMS key destroyed' findings at no additional cost. For the monthly IAM permission drift check, script `aws iam simulate-principal-policy` against a known-good permission baseline stored in version control, diffing the output to catch any newly introduced cloudtrail:StopLogging or kms:ScheduleKeyDeletion grants. Schedule this as a monthly GitHub Actions workflow triggered on the first of each month.

Evidence: Post-incident lessons-learned documentation should capture the following threat-specific artifacts as evidence of detection gap scope: (1) the complete timeline of CloudTrail digest file gaps (from S3 digest prefix analysis) mapped against known attacker dwell time, to quantify exactly how many hours of audit log coverage were lost across each of the five technique categories; (2) a copy of the IAM policy diff showing which permissions existed pre-incident versus post-containment, as evidence for risk register entry and potential regulatory notification assessment; (3) AWS Config timeline exports for CloudTrail trail resources and KMS key resources showing the configuration states at incident time, which serve as the authoritative secondary record when CloudTrail itself was the target of tampering.

Detection Guidance

Primary detection: Alert on the following API calls in CloudTrail: StopLogging, DeleteTrail, PutBucketLogging (with LoggingEnabled set to false), ScheduleKeyDeletion, DisableKey targeting KMS keys associated with CloudTrail or S3 log buckets. In Google Cloud Audit Logs, alert on logging.sinks.delete, logging.sinks.update (disabling a sink), storage.buckets.delete on log-destination buckets, and cloudkms.cryptoKeyVersions.destroy on CMEK keys protecting log storage. Secondary detection: Monitor CloudTrail delivery failure notifications (SNS topic alerts on delivery errors); an attacker who corrupts the KMS key will trigger delivery failures before you see the key-deletion event. Tertiary detection: Alert on IAM policy changes that add cloudtrail:StopLogging, cloudtrail:DeleteTrail, or equivalent GCP permissions to any principal that did not previously hold them.

Behavioral indicator: any of these events occurring outside a documented change-management window should be treated as a confirmed evasion attempt, not a configuration error. Apply NIST AU-6 (Audit Record Review, Analysis, and Reporting) cadence to review these alert classes at minimum daily.

Framework Mappings

MITRE-ATTACK

- **T1078.004** — Cloud Accounts
- **T1552.005** — Cloud Instance Metadata API
- **T1490** — Inhibit System Recovery
- **T1046** — Network Service Discovery
- **T1530** — Data from Cloud Storage
- **T1565.001** — Stored Data Manipulation
- **T1485** — Data Destruction
- **T1562.008** — Disable or Modify Cloud Logs
- **T1562.001** — Disable or Modify Tools

NIST-800-53R5

- **CP-9** — System Backup
- **CP-10** — System Recovery and Reconstitution
- **AC-3** — Access Enforcement
- **AC-6** — Least Privilege
- **SI-7** — Software, Firmware, and Information Integrity
- **SC-13** — Cryptographic Protection
- **SI-4** — System Monitoring

OWASP-TOP10-2021

- **A01:2021** — Broken Access Control
- **A08:2021** — Software and Data Integrity Failures

CIS-V8

- **3.3** — Configure Data Access Control Lists
- **2.5** — Allowlist Authorized Software
- **6.1** — Establish an Access Granting Process
- **6.2** — Establish an Access Revoking Process
- **8.2** — Collect Audit Logs

SOC2-TSC

- **CC6.1** — The entity implements logical access security software, infrastructure, and architectures over protected information assets

HIPAA-SECURITY

- **164.312(a)(1)** — Access Control
- **164.312(e)(1)** — Transmission Security

ISO-27001-2022

- **A.8.24** — Use of cryptography
- **A.5.23** — Information security for use of cloud services

NIST-CSF-2

- **DE.CM-01** — Networks and network services are monitored

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1078.004	Cloud Accounts	Defense-Evasion
T1552.005	Cloud Instance Metadata API	Credential-Access
T1490	Inhibit System Recovery	Impact
T1046	Network Service Discovery	Discovery
T1530	Data from Cloud Storage	Collection
T1565.001	Stored Data Manipulation	Impact
T1485	Data Destruction	Impact
T1562.008	Disable or Modify Cloud Logs	Defense-Evasion
T1562.001	Disable or Modify Tools	Defense-Evasion

Sources

Source	URL	Tier
Unit 42	https://unit42.paloaltonetworks.com/cloud-logging-defense-evasion/	T3
	https://unit42.paloaltonetworks.com/cloud-logging-defense-evasion/	T3
	https://unit42.paloaltonetworks.com/autonomous-ai-cloud-attacks/	T3
Collect AWS Key Management Service logs	https://docs.cloud.google.com/chronicle/docs/ingestion/default-pars...	T3
Ensure CloudTrail logs are encrypted at rest using KMS CMKs	https://orca.security/resources/blog/ensure-cloudtrail-logs-are-enc...	T3

DISCLAIMER

This intelligence report is produced by Tech Jacks Solutions Security Command Center (SCC) for informational purposes only. It does not constitute professional security advice, legal counsel, or an incident response engagement. The information herein is derived from publicly available sources and AI-assisted analysis; while every effort is made to ensure accuracy, Tech Jacks Solutions makes no warranties regarding completeness or timeliness. Organizations should conduct their own validation and consult qualified security professionals before taking action based on this report. Tech Jacks Solutions is not liable for any damages resulting from the use of this information.

Generated 2026-06-10 07:23 UTC by TJS Security Command Center