INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-06-22 19:03 UTC

Cloud Bucket Hijacking via Global Namespace Collision Enables Silent Data Stream Takeover Across AWS, GCP, and Azure

SECURITY ANALYSIS | HIGH | CVSS 7.5

SCC Item ID	SCC-STY-2026-0251
Type	Security Analysis
Severity	HIGH
CVSS Base Score	7.5
Affected Products	AWS S3, Amazon Data Firehose, Google Cloud Storage, Google Cloud Logging, Google Pub/Sub, Google Storage Transfer Service, Microsoft Azure (cross-subscription storage scenarios)
Published	2026-06-22T22:00:04+00:00
Discovery Source	Rss:T1 Threatintel

Executive Summary

Unit 42 researchers have disclosed a systemic architectural flaw across AWS, GCP, and Azure: when a cloud storage bucket is deleted, its globally unique name becomes available for re-registration by anyone, including attackers. An adversary with only delete-level permissions can claim the freed name and silently intercept audit logs, telemetry streams, and sensitive data pipelines originally pointed at the legitimate bucket, all without touching IAM policies or modifying any pipeline configuration. This technique exposes a blind spot in how enterprises assume immutability of cloud data destinations, and signals that namespace management is now a first-class security concern alongside identity and access controls.

Technical Analysis

The attack, disclosed by Palo Alto Networks Unit 42, exploits a property fundamental to how AWS S3, Google Cloud Storage, and Azure handle bucket naming: bucket names are globally unique and, once deleted, re-enter a first-come-first-served pool. The attack chain is deceptively simple. An adversary with permissions sufficient only to delete a target bucket, well below the threshold of modifying a logging sink, Pub/Sub subscription, or Firehose delivery stream, deletes it and immediately registers the same name under an attacker-controlled account. From that point forward, every service that was configured to write to the original bucket writes instead to the attacker's bucket. No pipeline configuration changes. No IAM policy modifications. No alerts fire because the destination name is identical.

The affected data pipelines are high-value by nature. Amazon Data Firehose, Google Cloud Logging sinks, Google Pub/Sub, and Google Storage Transfer Service are all commonly used to move audit logs, security telemetry, compliance records, and bulk data. Redirecting these streams means an attacker receives a continuous, real-time copy of an organization's security posture, including the very logs that would normally detect the intrusion.

The MITRE ATT&CK mapping is instructive. T1562.008 (Impair Defenses: Disable Cloud Logs) is the most critical: the victim organization believes logging is functioning normally while all records flow to the attacker. T1537 (Transfer Data to Cloud Account) describes the ongoing exfiltration mechanic. T1530 (Data from Cloud Storage) captures the passive collection enabled once the bucket is claimed. T1078.004 (Valid Accounts: Cloud Accounts) reflects the requirement that the attacker must possess a valid cloud account to register the freed bucket name. The technique also touches T1548 (Abuse Elevation Control Mechanism) because it achieves a capability, log redirection, that would normally require elevated IAM permissions, using only delete access.

Unit 42 noted no active exploitation at time of disclosure, but the attack surface is broad and the prerequisites are low. Any principal with bucket delete rights, including insider threats, compromised service accounts, or supply-chain-compromised CI/CD pipelines, could execute this in seconds. The attack persists indefinitely and silently until someone audits the ownership of the bucket receiving the data streams, a check that most cloud monitoring pipelines do not perform.

The cross-cloud dimension matters. Organizations running multi-cloud data pipelines or performing cloud-to-cloud migrations using services like Google Storage Transfer Service or AWS DataSync are exposed on multiple surfaces simultaneously. A single deleted bucket name claimed across providers could redirect multiple independent telemetry streams.

Action Checklist

1. Step 1: Assess exposure, audit every cloud data pipeline endpoint (S3 buckets, GCS buckets, Azure storage accounts) used as destinations for audit logs, SIEM forwarding, Firehose streams, Pub/Sub sinks, and Transfer Service jobs across AWS, GCP, and Azure. Confirm each destination bucket still exists and is owned by your organization.
2. Step 2: Verify bucket ownership continuously, implement a recurring automated check (daily or more frequent) that confirms the ownership account of every bucket receiving sensitive data streams. A bucket name resolving does not mean your account owns it; validate account ID against expected values. Map to NIST AC-3 (Access Enforcement) and CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory).
3. Step 3: Enforce bucket deletion protection, enable Object Lock (AWS S3), Retention Policies (GCS), and deletion prevention policies on all buckets serving as log or telemetry sinks. Where available, use cloud-provider controls to require MFA or secondary approval for bucket deletion. Aligns with NIST AC-6 (Least Privilege) and CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts).
4. Step 4: Scope delete permissions strictly, audit IAM roles and service accounts for bucket delete permissions. Remove delete rights from any principal that does not operationally require them. Logging pipeline service accounts should have write-only access, not delete. Reference NIST AC-6 (Least Privilege), CIS 6.1 (Establish an Access Granting Process), and D3-UAP (User Account Permissions).
5. Step 5: Audit logging sink integrity, verify that Google Cloud Logging sinks, Firehose delivery targets, and Pub/Sub subscription destinations are writing to expected, organization-owned storage and that no log volume anomaly (sudden drop in ingested logs) has occurred. Reference NIST AU-9 (Protection of

Audit Information) and CIS 8.2 (Collect Audit Logs).

6. Step 6: Update threat model, add namespace collision via bucket deletion to your cloud threat register as a low-privilege, high-impact lateral exfiltration path. Tag to MITRE T1562.008, T1537, and T1530. Flag any data pipeline destination as a potential pivot point for future adversary access to audit streams.

7. Step 7: Communicate findings, brief cloud infrastructure and SecOps leadership with specific context: this is not a permissions misconfiguration but an architectural behavior of all three major clouds. Generic cloud hardening guidance does not cover this. Decisions about deletion protection policies require cloud operations buy-in.

8. Step 8: Monitor developments, track Unit 42 and cloud provider security bulletins for follow-up mitigations, provider-side controls, or evidence of active exploitation. AWS, Google, and Azure have each acknowledged the research; watch for native bucket reclamation protections or namespace reservation features.

IR / Forensic Enrichment

Triage Priority	URGENT
Escalation Criteria	Escalate immediately to cloud infrastructure ownership and legal/compliance if Step 1 or Step 2 identifies any pipeline destination bucket where the confirmed account owner ID does not match your organization, or if Step 5 reveals an unexplained log volume drop coinciding with a known or suspected bucket deletion event, as either condition indicates a probable completed namespace collision and potential data exposure triggering breach notification obligations for any PII, PHI, or regulated data transiting the affected pipeline.
Recovery Notes	After containment actions (deletion protection enabled, delete permissions revoked), verify pipeline integrity by confirming that all Firehose streams, GCS logging sinks, Pub/Sub subscriptions, and Transfer Service jobs are actively delivering to buckets whose ownership account IDs have been re-validated against your inventory baseline. Monitor ingested log volume metrics for a minimum of 30 days post-containment to detect any recurrence or latent diversion not captured in the initial audit. If any pipeline was confirmed to have delivered data to an adversary-controlled bucket, treat all audit logs ingested through that pipeline for the exposure window as potentially tampered and reconstruct the audit record from secondary sources (CloudTrail, GCS Admin Activity logs, Azure Activity Log) that write to independently protected destinations.
Forensic Artifacts	AWS CloudTrail `DeleteBucket` events (s3.amazonaws.com) with requestor principal ARN, timestamp, and bucket name — cross-referenced against the pipeline destination inventory to identify the exact deletion event that opened the namespace collision window | GCS Admin Activity audit log entries for `storage.buckets.delete` and `storage.buckets.create` events on the same bucket name within a compressed timeframe, surfacing the adversary re-registration window | AWS Firehose CloudWatch metrics `DeliveryToS3.Records` and `DeliveryToS3.DataFreshness` time-series data showing the inflection point where log volume to a suspect destination bucket changed, establishing the earliest possible time of active data interception | IAM credential report and access key last-used timestamps for any service account or role holding `s3:DeleteBucket` or GCS `storage.buckets.delete` permissions on pipeline destination buckets — identifies whether the deletion was performed by a legitimate principal or a compromised credential | GCS Logging sink export configuration snapshots (`gcloud logging sinks list --format=json`) and Pub/Sub subscription metadata captured at the time of investigation, preserving the pipeline destination state as it existed before any remediation altered it

Per-Action IR Details

Step 1: Assess exposure — audit every cloud data pipeline endpoint (S3 buckets, GCS buckets, Azure storage accounts) used as destinations for audit logs, SIEM forwarding, Firehose streams, Pub/Sub sinks, and Transfer Service jobs across AWS, GCP, and Azure. Confirm each destination bucket still exists and is owned by your organization.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: establish scope of exposure by inventorying all pipeline destinations before determining whether namespace collision has already occurred

Controls: NIST AC-2 (Account Management), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory)

Compensating: Run AWS CLI: `aws s3api get-bucket-location --bucket ` and `aws s3api get-bucket-acl --bucket ` for each S3 destination; verify the owner ID matches your AWS account ID. For GCS: `gcloud storage buckets describe gs:// --format='value(owner)'`. For Azure: `az storage account show --name --query id`. A 2-person team can script these as a single-pass shell loop against an exported pipeline destination list.

Evidence: Before any remediation action, export current pipeline configuration state as volatile evidence: AWS Firehose delivery stream configurations (`aws firehose describe-delivery-stream`), GCS Logging sink descriptors (`gcloud logging sinks list --format=json`), Pub/Sub subscription metadata (`gcloud pubsub subscriptions describe`), and Azure diagnostic settings (`az monitor diagnostic-settings list`). These configurations may be altered by an adversary post-hijack and represent the pre-action ground truth.

Step 2: Verify bucket ownership continuously — implement a recurring automated check (daily or more frequent) that confirms the ownership account of every bucket receiving sensitive data streams. A bucket name resolving does not mean your account owns it; validate account ID against expected values. Map to NIST AC-3 (Access Enforcement) and CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory).

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: continuous monitoring to detect namespace collision events where a hijacked bucket name resolves successfully but ownership has silently changed

Controls: NIST AC-3 (Access Enforcement), NIST AU-6 (Audit Record Review, Analysis, and Reporting), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory)

Compensating: Schedule a daily cron job that runs `aws s3api get-bucket-acl --bucket | jq '.Owner.ID'` for every S3 pipeline destination and diffs the output against a baseline file of expected canonical user IDs. For GCS, use `gcloud storage buckets get-iam-policy gs://` and confirm the owning project number. Alert on any mismatch via email or webhook to a free Slack workspace. Achievable in under 50 lines of bash.

Evidence: Capture and retain daily ownership validation output as timestamped log files; these records establish the exact moment an ownership discrepancy first appeared, which is the forensic indicator of a completed namespace collision. Also capture CloudTrail `DeleteBucket` events and GCS Admin Activity audit log `storage.buckets.delete` entries for the relevant bucket names — the gap between deletion timestamp and adversary re-registration timestamp is critical for timeline reconstruction.

Step 3: Enforce bucket deletion protection — enable object lock, versioning hold, and deletion prevention policies on all buckets serving as log or telemetry sinks. Where available, use cloud-provider controls to require MFA or secondary approval for bucket deletion. Aligns with NIST AC-6 (Least Privilege) and CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts).

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment: close the deletion vector that enables namespace collision by preventing future bucket deletion from log and telemetry sink buckets without elevated authorization

Controls: NIST AC-6 (Least Privilege), CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts)

Compensating: For AWS: apply an S3 bucket policy with a `Deny` on `s3:DeleteBucket` for all principals except a dedicated break-glass IAM role, and enable S3 Object Lock in Compliance mode on log buckets via `aws s3api

put-object-lock-configuration`. For GCS: apply a `storage.buckets.delete` deny IAM condition at the project or folder level. For Azure: apply a resource lock (`az lock create --lock-type CanNotDelete`) on storage accounts receiving diagnostic logs. All steps are zero-cost using native provider CLI.

Evidence: Before enabling deletion protection, capture the current bucket deletion audit history: AWS CloudTrail `s3.amazonaws.com` events filtered for `DeleteBucket` across all regions for the past 90 days; GCS Admin Activity logs for `storage.buckets.delete`; Azure Activity Log filtered for `Microsoft.Storage/storageAccounts/delete`. This volatile historical record establishes whether a prior deletion-and-reclaim event has already occurred, which would indicate active compromise rather than residual risk.

Step 4: Scope delete permissions strictly — audit IAM roles and service accounts for bucket delete permissions. Remove delete rights from any principal that does not operationally require them. Logging pipeline service accounts should have write-only access, not delete. Reference NIST AC-6 (Least Privilege), CIS 6.1 (Establish an Access Granting Process), and D3-UAP (User Account Permissions).

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment: remove the low-privilege delete permission that is the sole prerequisite for a namespace collision attack against log sink buckets

Controls: NIST AC-6 (Least Privilege), NIST AC-2 (Account Management), CIS 6.1 (Establish an Access Granting Process), CIS 6.2 (Establish an Access Revoking Process)

Compensating: For AWS: run `aws iam list-entities-for-policy --policy-arn arn:aws:iam::aws:policy/AmazonS3FullAccess` and cross-reference against Firehose and logging service roles; revoke `s3:DeleteBucket` and `s3:DeleteObject` from all pipeline service accounts using inline deny policies. For GCS: `gcloud projects get-iam-policy --format=json | jq '.bindings[] | select(.role | contains("storage.admin"))'` identifies overprivileged service accounts on logging buckets. Replace with the custom role containing only `storage.objects.create`.

Evidence: Before revoking any IAM permissions, export and preserve the complete current IAM state: `aws iam generate-credential-report` and `aws iam get-account-authorization-details` for AWS; `gcloud projects get-iam-policy` for GCS; `az role assignment list` for Azure. Revoking permissions is a live-state change — the pre-revocation IAM configuration is volatile forensic evidence needed to determine whether a principal with delete rights was exploited or abused prior to this response action.

Step 5: Audit logging sink integrity — verify that Google Cloud Logging sinks, Firehose delivery targets, and Pub/Sub subscription destinations are writing to expected, organization-owned storage and that no log volume anomaly (sudden drop in ingested logs) has occurred. Reference NIST AU-9 (Protection of Audit Information) and CIS 8.2 (Collect Audit Logs).

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: identify evidence of active or historical log stream diversion by detecting volume anomalies and mismatched sink destinations, which are the primary observable indicators of a completed namespace collision attack

Controls: NIST AU-9 (Protection of Audit Information), NIST AU-6 (Audit Record Review, Analysis, and Reporting), CIS 8.2 (Collect Audit Logs)

Compensating: For GCS sinks: run `gcloud logging sinks list --format=json` and compare `destination` field values against your asset inventory baseline. For Firehose: `aws firehose list-delivery-streams` then `describe-delivery-stream` for each, checking `S3DestinationDescription.BucketARN` account ID prefix against expected account. For log volume anomaly detection without a SIEM: query Firehose CloudWatch metrics `DeliveryToS3.Records` and `DeliveryToS3.DataFreshness` for a sudden drop; in GCS use `gcloud logging metrics` or export sink metrics to Cloud Monitoring and compare 7-day rolling average against current ingestion rate.

Evidence: The primary forensic indicator of a completed namespace collision against a log sink is a sudden, unexplained drop in ingested log volume with no corresponding decrease in source activity — capture time-series log ingestion metrics for all sinks covering the maximum retention window available before any reconfiguration. Additionally, if a Firehose or Pub/Sub stream was actively delivering to a hijacked bucket, the adversary-controlled bucket's access logs (if retrievable) would contain records of your organization's telemetry writes; request any available

provider-side evidence of cross-account write activity to the suspect bucket name.

Step 6: Update threat model — add namespace collision via bucket deletion to your cloud threat register as a low-privilege, high-impact lateral exfiltration path. Tag to MITRE T1562.008, T1537, and T1530. Flag any data pipeline destination as a potential pivot point for future adversary access to audit streams.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: update threat model and risk register with the namespace collision attack class to improve future preparation and detection capability

Controls: NIST AU-2 (Event Logging)

Compensating: Document the threat pattern in a markdown-formatted threat register entry (stored in version control) that describes the attack preconditions (delete permission on a pipeline-destination bucket), the exploitation mechanism (re-registration of freed global namespace), observable indicators (ownership validation mismatch, log volume drop), and detection gaps this exposes. Assign a risk owner from cloud operations and set a quarterly review cadence. No tooling cost required.

Evidence: No volatile state is altered by this step; evidence capture is not a prerequisite. However, document any IOCs identified during Steps 1–5 — specifically: bucket names where ownership mismatches were detected, timestamps of `DeleteBucket` events cross-referenced with adversary re-registration windows, and any log volume anomaly time windows — as these feed directly into the threat register entry and future detection rule development.

Step 7: Communicate findings — brief cloud infrastructure and SecOps leadership with specific context: this is not a permissions misconfiguration but an architectural behavior of all three major clouds. Generic cloud hardening guidance does not cover this. Decisions about deletion protection policies require cloud operations buy-in.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: communicate lessons learned and architectural risk findings to stakeholders required to authorize structural changes to cloud pipeline configurations

Compensating: Prepare a one-page executive brief using the Unit 42 research disclosure as the authoritative source reference. Structure it as: (1) what the architectural behavior is and why it is not addressable by standard IAM hardening, (2) which specific pipelines in your environment were confirmed exposed during Step 1, (3) what containment actions have been taken, (4) what decisions require cloud operations authorization (deletion protection policies, break-glass IAM roles). No tooling required; a 2-person team can produce this in under 2 hours.

Evidence: No volatile state is altered by this step. Attach the output of Steps 1–2 (pipeline destination audit results and ownership validation findings) and Step 5 (log volume anomaly analysis) as supporting evidence in the leadership brief to ground the architectural risk discussion in organization-specific exposure data rather than generic vendor advisories.

Step 8: Monitor developments — track Unit 42 and cloud provider security bulletins for follow-up mitigations, provider-side controls, or evidence of active exploitation. AWS, Google, and Azure have each acknowledged the research; watch for native bucket reclamation protections or namespace reservation features.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: maintain situational awareness on a systemic architectural vulnerability where provider-side mitigations are pending and exploitation status may change

Controls: CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: Subscribe to AWS Security Bulletins (https://aws.amazon.com/security/security-bulletins/), Google Cloud Security Advisories (https://cloud.google.com/support/bulletins), and Azure Security Updates (https://msrc.microsoft.com/update-guide/) via RSS or email. Set a calendar-based review cadence of no less than weekly until each provider publishes a native namespace reservation or bucket reclamation control. Track the Unit 42 Threat Research blog directly for follow-up publication. A 2-person team can manage this with a shared tracking document and provider RSS feeds at zero cost.

Evidence: No volatile state is altered by this step. Maintain a running log of provider bulletin publications and dates as part of the threat register entry created in Step 6; this creates an auditable record of when provider-side mitigations became available relative to your exposure window, which may be relevant for regulatory reporting if active exploitation is later confirmed.

Detection Guidance

Detection for this technique is difficult precisely because no configuration changes occur. Focus on three detection surfaces:

1. Log volume anomalies: A sudden and unexplained drop in log ingestion to a SIEM or security data lake is the most likely observable signal. If a Firehose stream, Logging sink, or Pub/Sub subscription stops delivering expected volume but the pipeline configuration appears intact, treat it as a potential namespace collision event. Monitor against NIST AU-5 (Response to Audit Logging Process Failures) and alert on delivery gaps. CIS 8.2 (Collect Audit Logs) requires verifying that logging remains active across the enterprise.

2. Bucket ownership validation: Implement checks that verify the AWS Account ID or GCP project number associated with the destination bucket for each critical pipeline, not just the bucket name. A name collision will show the correct bucket name but a different owning account. Tools like AWS S3 HeadBucket with account-ownership assertions or GCP storage bucket IAM get calls from a monitoring service account can surface this. Apply D3-SFA (System File Analysis) principles adapted to cloud storage metadata.

3. IAM delete activity on sink buckets: Alert on any DeleteBucket, storage.buckets.delete, or equivalent API call against buckets designated as log or telemetry sinks. These buckets should almost never be deleted. Log these calls to CloudTrail (AWS), Cloud Audit Logs (GCP), or Azure Monitor, then alert immediately. Reference NIST AU-2 (Event Logging) and AU-6 (Audit Record Review, Analysis, and Reporting).

4. Hunting hypothesis: Query your CloudTrail or GCP Admin Activity logs for DeleteBucket events on any bucket that appears in a Firehose configuration, Logging sink, Pub/Sub subscription, or Transfer Service job within the preceding 90 days (or your organization's log retention window, whichever is longer). Cross-reference against current pipeline configurations to identify any pipeline pointing to a bucket that was deleted and has since re-appeared.

Indicators of Compromise

Type	Value	Context	Confidence
URL	`Pending — refer to Unit 42 (https://unit42.paloaltonetworks.com/cloud-bucket-hijacking-risks/) for published indicators`	Unit 42's disclosure may include attacker-controlled bucket name patterns, proof-of-concept account identifiers, or API call signatures associated with namespace collision testing; the source article should be reviewed directly for any published indicators	LOW

Framework Mappings

MITRE-ATTACK

- **T1562.008** — Disable or Modify Cloud Logs
- **T1537** — Transfer Data to Cloud Account
- **T1078.004** — Cloud Accounts
- **T1548** — Abuse Elevation Control Mechanism
- **T1530** — Data from Cloud Storage

NIST-800-53R5

- **AC-6** — Least Privilege
- **CM-6** — Configuration Settings
- **AC-3** — Access Enforcement
- **SI-4** — System Monitoring
- **IR-5** — Incident Monitoring

OWASP-TOP10-2021

- **A01:2021** — Broken Access Control

CIS-V8

- **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

NIST-CSF-2

- **DE.CM-01** — Networks and network services are monitored
- **DE.AE-08** — Incidents are declared when adverse events meet the defined incident criteria

ISO-27001-2022

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

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1562.008	Disable or Modify Cloud Logs	Defense-Evasion
T1537	Transfer Data to Cloud Account	Exfiltration
T1078.004	Cloud Accounts	Defense-Evasion

Technique ID	Technique Name	Tactic
T1548	Abuse Elevation Control Mechanism	Privilege-Escalation
T1530	Data from Cloud Storage	Collection

Sources

Source	URL	Tier
Unit 42	https://unit42.paloaltonetworks.com/cloud-bucket-hijacking-risks/	T3
	https://unit42.paloaltonetworks.com/cloud-bucket-hijacking-risks/	T3
Transfer from Amazon S3 to Cloud Storage	https://docs.cloud.google.com/storage-transfer/docs/create-transfer...	T3
Migrating Google Cloud Storage to Amazon S3 using AWS DataSync	https://aws.amazon.com/blogs/storage/migrating-google-cloud-storage...	T3
Migrate private s3 hosted objects to gcs - Google Developer forums	https://discuss.google.dev/t/migrate-private-s3-hosted-objects-to-g...	T3

Generated 2026-06-22 19:03 UTC by TJS Security Command Center