INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-06-17 07:17 UTC

Vertex AI SDK Bucket Squatting Enables Cross-Tenant RCE via Pickle Deserialization

CVE VULNERABILITY | HIGH | CVSS 7.5

SCC Item ID	SCC-CVE-2026-0314
Type	CVE Vulnerability
Severity	HIGH
CVSS Base Score	7.5
Affected Products	Google Cloud Vertex AI Python SDK (google-cloud-aiplatform versions 1.139.0-1.140.0); fixed in v1.148.0 (released April 15, 2026)
Published	2026-06-16T10:00:29+00:00
Discovery Source	Rss:T1 Threatintel

Executive Summary

A vulnerability in Google Cloud's Vertex AI Python SDK (versions 1.139.0-1.140.0) allowed an unauthenticated attacker to hijack AI model uploads by claiming a predictable staging storage bucket before the victim's upload occurred. Once claimed, the attacker could poison the bucket with a malicious payload, triggering remote code execution inside the victim's model serving infrastructure when the model loaded. Organizations using the affected SDK versions to upload or serve AI models in Google Cloud are directly exposed until they upgrade to v1.148.0.

Technical Analysis

The Vertex AI Python SDK (google-cloud-aiplatform v1.139.0-1.140.0) generated deterministic, guessable names for staging Google Cloud Storage buckets used during model uploads. An unauthenticated attacker could preemptively register a bucket matching that naming scheme, a technique called bucket squatting, before the victim's SDK created it. The SDK performed no bucket ownership validation, so it wrote model artifacts to the attacker-controlled bucket without error. The attacker then poisoned the bucket with a maliciously crafted pickle-serialized object. When the victim's Vertex AI serving environment loaded the model and deserialized the payload, the attacker achieved remote code execution (RCE) in the victim's infrastructure. No authentication to the victim's GCP project was required. The attack chain combines CWE-340 (predictable identifier generation), CWE-502 (deserialization of untrusted data), CWE-284 (improper access control), and CWE-362 (race condition via shared resource). Relevant MITRE ATT&CK techniques include T1190 (Exploit Public-Facing Application), T1059.006 (Python scripting), T1530 (Data from Cloud Storage), T1195.001 (Supply Chain Compromise), and T1567.002 (Exfiltration to Cloud Storage). No CVE identifier has been assigned as of the publication date of this

advisory. A CVE may be assigned retrospectively by NVD or Google's security advisory process. Google patched the issue in v1.148.0, released April 15, 2026. Source: Unit 42 (Palo Alto Networks), https://unit42.paloaltonetworks.com/hijacking-vertex-ai-model/

Action Checklist

1. Step 1: Containment. Immediately identify all systems and pipelines running google-cloud-aiplatform v1.139.0 or v1.140.0. Suspend model upload jobs using these versions until the upgrade is complete. Audit all GCS staging buckets created by the SDK during the exposure window (between the first deployment of v1.139.0 and the upgrade to v1.148.0). The affected SDK versions generate staging buckets with predictable names; consult the Vertex AI SDK source or Google's advisory for the exact naming convention. Flag any bucket matching that pattern owned by an account outside your GCP organization or created by an external principal.
2. Step 2: Detection. Query your GCS audit logs (Cloud Audit Logs: Data Access logs) for staging bucket creation events matching the SDK's deterministic naming pattern during the exposure window. Look for bucket creation events originating from accounts outside your organization, or bucket access events by the SDK to buckets not owned by your project. Review Vertex AI model serving logs for unexpected Python subprocess spawning or anomalous outbound network connections post-model-load, which may indicate pickle payload execution. NIST AU-6 and CIS 8.2 apply; ensure audit log collection was enabled across GCS and Vertex AI during this period.
3. Step 3: Eradication. Upgrade google-cloud-aiplatform to v1.148.0 or later via pip: 'pip install --upgrade google-cloud-aiplatform'. Verify the installed version with 'pip show google-cloud-aiplatform'. Delete or transfer ownership of any staging buckets created by the SDK during the affected version window. Do not reuse model artifacts loaded from staging buckets during the exposure period; treat them as potentially compromised.
4. Step 4: Recovery. After upgrading, re-upload all models that were staged using v1.139.0 or v1.140.0, sourcing artifacts from a verified, internal repository rather than previously staged buckets. Validate bucket ownership for all new staging buckets created post-upgrade. Monitor Vertex AI serving infrastructure for anomalous process execution and unexpected outbound connections for at least 30 days post-remediation, per NIST SI-4 (system monitoring) principles. Confirm audit logging (NIST AU-2, CIS 8.2) is active on all GCS buckets and Vertex AI endpoints.
5. Step 5: Post-Incident. Conduct a review of all SDK and library dependencies used in ML pipelines for similar predictable resource naming patterns. Implement a policy requiring pickle deserialization to be replaced with safer serialization formats (e.g., ONNX, SavedModel) in model serving pipelines, addressing the CWE-502 root cause. Apply least privilege (NIST AC-6, CIS 5.4) to GCS staging bucket access so SDK service accounts can only read from project-owned buckets. Add bucket ownership validation to pre-upload SDK integration tests. Reference D3-SFA (System File Analysis) for ongoing model artifact integrity monitoring.

IR / Forensic Enrichment

Triage Priority	URGENT

Escalation Criteria	Escalate immediately to CISO and legal/privacy counsel if Cloud Audit Logs confirm a staging bucket was created by an external principal and subsequently accessed by the Vertex AI SDK (indicating successful bucket squatting), or if Vertex AI serving logs show subprocess spawning or unexpected outbound connections post-model-load, as either condition indicates confirmed cross-tenant RCE with potential data exfiltration from model serving infrastructure that may trigger breach notification obligations.
Recovery Notes	Re-upload all models staged during the v1.139.0–v1.140.0 exposure window exclusively from verified internal artifact repositories using the patched v1.148.0 SDK, and confirm SHA-256 integrity of each artifact before serving. Validate GCS IAM policies on all new staging buckets immediately post-upload to confirm project ownership, and enforce this check as a blocking CI/CD gate going forward. Maintain enhanced monitoring of Vertex AI serving endpoint logs and VPC Flow Logs for anomalous subprocess execution and unexpected outbound connections for a minimum of 30 days, given the risk of delayed or persistent pickle payload effects in long-running serving containers.
Forensic Artifacts	GCS Cloud Audit Logs (Data Access) — 'storage.buckets.create' and 'storage.objects.create' events during the exposure window, filtered for principalEmail values outside your organization's domain, which would confirm external bucket squatting of the SDK's deterministic staging bucket names | GCS staging bucket IAM policy snapshots ('gsutil iam get gs://BUCKET_NAME') — ACL entries showing unexpected ownerProject or allUsers/allAuthenticatedUsers grants on buckets matching the SDK's staging naming pattern are direct evidence of bucket hijacking | Forensic copies of model artifact files retrieved from suspect staging buckets — pickle files (.pkl, .pickle) or files with Python pickle magic bytes (protocol 2: '\x80\x02', protocol 5: '\x80\x05') containing 'cos\nsystem' or 'csubprocess\ncall' opcodes are direct evidence of a malicious payload planted by the attacker | Vertex AI model serving endpoint stdout/stderr logs — entries showing Python 'subprocess', 'os.system', 'exec', or 'eval' calls occurring within the model loading lifecycle (not attributable to legitimate serving code) indicate pickle payload execution triggered by loading a poisoned model artifact | VPC Flow Logs from Vertex AI serving infrastructure — unexpected outbound TCP connections to external IP addresses originating from the serving container within 60 seconds of a model load event, particularly on non-standard ports, are consistent with C2 callback or data exfiltration from a successfully executed pickle payload

Per-Action IR Details

Step 1: Containment — Immediately identify all systems and pipelines running google-cloud-aiplatform v1.139.0 or v1.140.0. Suspend model upload jobs using these versions until the upgrade is complete. Audit all GCS staging buckets created by the SDK during the exposure window (between the first deployment of v1.139.0 and the upgrade to v1.148.0) for unexpected ownership or modified contents.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment Strategy: isolate affected components to prevent further staging bucket poisoning or additional pickle payload delivery to Vertex AI serving infrastructure

Controls: NIST AC-6 (Least Privilege), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), CIS 2.1 (Establish and Maintain a Software Inventory)

Compensating: Run 'pip list | grep google-cloud-aiplatform' across all ML pipeline hosts or CI/CD runner images. For GCS bucket ownership audit without enterprise tooling, use 'gsutil ls -L gs://BUCKET_NAME' to inspect bucket metadata and ACLs; pipe output to a file for comparison against expected project IDs. Use a short bash loop over known staging bucket name prefixes derived from the SDK's deterministic naming convention to enumerate and check ownership: 'for b in $(gsutil ls); do gsutil iam get $b; done > bucket_iam_audit.txt'.

Evidence: Before suspending any upload jobs or modifying bucket permissions, capture: (1) current output of 'gsutil ls -r gs://' scoped to SDK staging prefixes to record bucket inventory at time of discovery; (2) 'gsutil iam get

gs://BUCKET_NAME' output for each staging bucket in the exposure window to document current ACL state; (3) Cloud Audit Logs — Data Access log export for all GCS bucket creation and object write events during the exposure window (v1.139.0 first deploy through present), preserving log timestamps and principal identifiers before any bucket deletion or ownership transfer alters the record.

Step 2: Detection — Query your GCS audit logs (Cloud Audit Logs: Data Access logs) for staging bucket creation events matching the SDK's deterministic naming pattern during the exposure window. Look for bucket creation events originating from accounts outside your organization, or bucket access events by the SDK to buckets not owned by your project. Review Vertex AI model serving logs for unexpected Python subprocess spawning or anomalous outbound network connections post-model-load, which may indicate pickle payload execution. NIST AU-6 and CIS 8.2 apply — ensure audit log collection was enabled across GCS and Vertex AI during this period.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: correlate GCS Data Access audit logs and Vertex AI serving logs to determine whether bucket squatting occurred and whether a pickle payload reached execution in any model serving container

Controls: NIST AU-6 (Audit Record Review, Analysis, And Reporting), NIST AU-3 (Content Of Audit Records), CIS 8.2 (Collect Audit Logs)

Compensating: Without a SIEM, export Cloud Audit Logs to BigQuery (free tier available) and run: 'SELECT protoPayload.authenticationInfo.principalEmail, protoPayload.resourceName, timestamp FROM your_project.cloudaudit_googleapis_com_data_access WHERE protoPayload.methodName="storage.buckets.create" AND timestamp BETWEEN "EXPOSURE_START" AND "EXPOSURE_END"' to surface bucket creation by external principals. For Vertex AI serving container subprocess detection without EDR, enable Cloud Logging on the serving endpoint and filter for log entries containing 'subprocess', 'os.system', 'exec', or unexpected outbound IP connections in the container stdout/stderr stream using: 'gcloud logging read "resource.type=ml_job AND textPayload:subprocess" --project=PROJECT_ID'.

Evidence: This step is analytical and does not alter live state, but preserve before proceeding to eradication: (1) full Cloud Audit Logs — Data Access export (JSON) for GCS and Vertex AI resources covering the entire exposure window, including 'storage.buckets.create', 'storage.objects.create', and 'storage.objects.get' method names; (2) Vertex AI model serving endpoint logs showing model load events and any Python runtime output, specifically filtering for pickle module invocations or unexpected child process creation; (3) network flow logs (VPC Flow Logs) from Vertex AI serving infrastructure for any outbound connections to external IPs occurring within 60 seconds of a model load event, which would indicate C2 callback from a successful pickle payload execution.

Step 3: Eradication — Upgrade google-cloud-aiplatform to v1.148.0 or later via pip: 'pip install --upgrade google-cloud-aiplatform'. Verify the installed version with 'pip show google-cloud-aiplatform'. Delete or transfer ownership of any staging buckets created by the SDK during the affected version window. Do not reuse model artifacts loaded from staging buckets during the exposure period — treat them as potentially compromised.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication: remove the vulnerable SDK version from all pipeline hosts and container images, eliminate attacker-controlled or potentially poisoned GCS staging buckets, and quarantine model artifacts that transited those buckets during the exposure window

Controls: NIST SI-2 (Flaw Remediation), CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management), CIS 2.2 (Ensure Authorized Software is Currently Supported)

Compensating: For containerized ML pipelines, rebuild all Docker images from base using a pinned 'google-cloud-aiplatform==1.148.0' in requirements.txt rather than patching running containers. Verify the upgrade in all images with 'docker run --rm IMAGE pip show google-cloud-aiplatform | grep Version'. For bucket deletion, use 'gsutil rm -r gs://BUCKET_NAME' after confirming artifact capture is complete. Maintain a manifest of deleted bucket names and deletion timestamps for the post-incident record.

Evidence: Before upgrading (which alters the installed package state) and before deleting staging buckets (which destroys potential evidence), capture: (1) complete 'pip freeze' output from each affected host or container image to document the full dependency tree at time of incident; (2) 'gsutil cp -r gs://SUSPECT_BUCKET ./forensic_copy/' for each staging bucket identified in Step 1 to preserve bucket contents including any malicious pickle files (handle in an isolated environment — do not execute); (3) hash (SHA-256) of all model artifact files retrieved from staging buckets during the exposure window using 'sha256sum MODEL_FILE' for later comparison against known-good artifacts from your internal model registry.

Step 4: Recovery — After upgrading, re-upload all models that were staged using v1.139.0 or v1.140.0, sourcing artifacts from a verified, internal repository rather than previously staged buckets. Validate bucket ownership for all new staging buckets created post-upgrade. Monitor Vertex AI serving infrastructure for anomalous process execution and unexpected outbound connections for at least 30 days post-remediation, per NIST SI-4 (system monitoring) principles. Confirm audit logging (NIST AU-2, CIS 8.2) is active on all GCS buckets and Vertex AI endpoints.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery: restore Vertex AI model serving to a verified clean state by re-staging models from trusted internal sources using the patched SDK, and confirm monitoring coverage to detect any persistence or delayed payload execution

Controls: NIST AU-2 (Event Logging), NIST AC-6 (Least Privilege), CIS 8.2 (Collect Audit Logs), CIS 4.6 (Securely Manage Enterprise Assets and Software)

Compensating: For teams without a formal internal model registry, establish a temporary verified source by storing model artifacts in a project-owned GCS bucket with object versioning enabled ('gsutil versioning set on gs://TRUSTED_BUCKET') and restricting write access to a single authorized service account via IAM. After re-upload with v1.148.0, validate new staging bucket ownership immediately with 'gsutil iam get gs://NEW_STAGING_BUCKET' and confirm the owning project matches your organization. Set up a daily cron job running 'gcloud logging read' queries against Vertex AI serving logs filtering for subprocess or exec patterns for the 30-day monitoring window.

Evidence: Before serving any re-uploaded model, verify: (1) SHA-256 hash of each re-uploaded model artifact matches the hash of the source file from your internal repository — document the comparison in writing; (2) 'gsutil stat gs://NEW_STAGING_BUCKET/MODEL_OBJECT' output confirming bucket ownership fields show your project ID, not an external principal; (3) a baseline snapshot of expected outbound network connections from the Vertex AI serving endpoint (VPC Flow Logs, first 24 hours post-recovery) to establish a normal behavior baseline for the 30-day monitoring period.

Step 5: Post-Incident — Conduct a review of all SDK and library dependencies used in ML pipelines for similar predictable resource naming patterns. Implement a policy requiring pickle deserialization to be replaced with safer serialization formats (e.g., ONNX, SavedModel) in model serving pipelines, addressing the CWE-502 root cause. Apply least privilege (NIST AC-6, CIS 5.4) to GCS staging bucket access so SDK service accounts can only read from project-owned buckets. Add bucket ownership validation to pre-upload SDK integration tests. Reference D3-SFA (System File Analysis) for ongoing model artifact integrity monitoring.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: conduct lessons-learned review, update ML pipeline security policies to prevent predictable resource squatting and unsafe deserialization, and improve detective controls for future Vertex AI SDK supply chain risks

Controls: NIST AC-6 (Least Privilege), NIST AU-6 (Audit Record Review, Analysis, And Reporting), CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: For the dependency review, run 'pip-audit' (free, open source) against all ML pipeline requirements files to surface known CVEs in google-cloud-aiplatform and dependencies. For bucket ownership validation in CI/CD integration tests without enterprise tooling, add a pytest fixture that calls 'subprocess.check_output(["gsutil", "iam", "get", staging_bucket])' post-upload and asserts the owning project matches the expected project ID — fail the pipeline if it does not. For D3-SFA model artifact integrity monitoring, implement a YARA rule scanning model files for Python

pickle opcodes ('\x80\x05' magic bytes for protocol 5, or 'cos\nsystem' patterns indicating OS command execution) run as a pre-serve gate in the model loading pipeline.

Evidence: This step does not alter active incident state, but produce and preserve: (1) a written inventory of all ML pipeline SDK dependencies with version pins and their evaluated resource naming behaviors, retained as a post-incident artifact; (2) documented IAM policy diff for GCS staging bucket service accounts showing the before/after least-privilege change; (3) the updated integration test results showing bucket ownership validation passing for v1.148.0 uploads, stored in version control as evidence of control implementation.

Detection Guidance

Enable and query GCS Data Access audit logs in Google Cloud Audit Logs for staging bucket creation events matching the SDK's predictable naming pattern. Flag any bucket created under the expected naming scheme that is owned by an account outside your GCP organization. Look for SDK write operations to buckets not listed in your project's bucket inventory. In Vertex AI serving logs, hunt for anomalous Python interpreter invocations, unexpected child process spawning, or outbound network connections initiated immediately after model load - these are behavioral indicators of pickle payload execution (T1059.006, T1190). Cross-reference GCS object write timestamps against your model upload job logs to detect writes occurring from sources other than your SDK service account. If you use a SIEM, create a rule alerting on GCS bucket IAM policy changes where the new owner is outside your organization's domain, correlated with Vertex AI model registration events. No public IOC signatures (hashes, IPs, domains) are available for this vulnerability at this time. NIST AU-6 and AU-12 govern the log review requirements; CIS 8.2 governs audit log collection. Source: Unit 42, https://unit42.paloaltonetworks.com/hijacking-vertex-ai-model/

Framework Mappings

MITRE-ATTACK

- **T1567.002** — Exfiltration to Cloud Storage
- **T1059.006** — Python
- **T1078.004** — Cloud Accounts
- **T1190** — Exploit Public-Facing Application
- **T1195.001** — Compromise Software Dependencies and Development Tools
- **T1650** — Acquire Access
- **T1530** — Data from Cloud Storage

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
- **AC-3** — Access Enforcement
- **SI-16** — Memory Protection

- **AT-2** — Literacy Training and Awareness

OWASP-TOP10-2021

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

CIS-V8

- **16.10** — Apply Secure Design Principles in Application Architectures
- **6.1** — Establish an Access Granting Process
- **6.2** — Establish an Access Revoking Process
- **7.3** — Perform Automated Operating System Patch Management
- **7.4** — Perform Automated Application Patch Management
- **14.2** — Train Workforce Members to Recognize Social Engineering Attacks

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

ISO-27001-2022

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

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1567.002	Exfiltration to Cloud Storage	Exfiltration
T1059.006	Python	Execution
T1078.004	Cloud Accounts	Defense-Evasion
T1190	Exploit Public-Facing Application	Initial-Access
T1195.001	Compromise Software Dependencies and Development Tools	Initial-Access
T1650	Acquire Access	Resource-Development
T1530	Data from Cloud Storage	Collection

Sources

Source	URL	Tier
Unit 42	https://unit42.paloaltonetworks.com/hijacking-vertex-ai-model/	T3
	https://unit42.paloaltonetworks.com/hijacking-vertex-ai-model/	T3
google-cloud-aiplatform - PyPI	https://pypi.org/project/google-cloud-aiplatform/	T3
Vertex AI SDK for Python | Python client libraries	https://docs.cloud.google.com/python/docs/reference/aiplatform/latest	T3
Releases · googleapis/python-aiplatform - GitHub	https://github.com/googleapis/python-aiplatform/releases	T3

Generated 2026-06-17 07:17 UTC by TJS Security Command Center