CSAI

cloud security alliance®

CSAI Foundation

Cloud Security Alliance AI Safety Initiative

Overprivileged by Design: AI Agents as Cloud Escalation Vectors

The Vertex AI P4SA Vulnerability Class and Cloud-Native AI Identity Risk

Unofficial AI-assisted Research

2026-04-02

Key Takeaways

- Palo Alto Networks Unit 42 disclosed on March 31, 2026, that the Per-Project, Per-Product Service Agent (P4SA) provisioned by default for Vertex AI Agent Engine deployments carries excessive permissions that allow a compromised or misconfigured AI agent to exfiltrate sensitive data, access private container registries, and pivot laterally across a Google Cloud project.
- The attack chain requires no novel exploit: it relies entirely on querying Google Cloud's standard metadata service to harvest pre-existing, over-scoped credentials — a technique available to any code executing within the agent runtime.
- This is a vulnerability class, not a single bug. The underlying pattern — cloud AI platforms provisioning broad default service accounts for agent execution contexts — applies wherever AI service agents inherit platform-level credentials without least-privilege scoping.
- Google has acknowledged the findings and updated official Vertex AI documentation to strongly recommend a Bring Your Own Service Account (BYOSA) architecture [3], which replaces the over-privileged default P4SA with an organization-controlled, minimally scoped service account.
- HiddenLayer's 2026 AI Threat Landscape Report, based on a survey of 250 IT and security leaders, found that one in eight reported AI breaches is now linked to agentic systems — a figure CSA expects will grow as enterprise agent deployments accelerate [1].
- Organizations deploying AI agents on any cloud-hosted AI platform should immediately audit the default service account permissions provisioned for their agent execution environments and enforce least-privilege boundaries as a prerequisite to production deployment.

Background

The deployment of AI agents on cloud infrastructure introduces a category of identity risk that conventional IAM governance frameworks were not designed to address. Traditional cloud workloads are generally expected to receive service accounts scoped to their functional requirements, though in practice this discipline is inconsistently applied across workload types. AI agents deployed through managed platforms like Vertex AI Agent Engine introduce a distinct dimension to this challenge: the

platform itself provisions a service agent on the organization's behalf, often with permissions that reflect the platform's operational breadth rather than the specific agent's functional needs. The result is a default permission posture that violates the principle of least privilege before the organization has written a single line of agent code.

Google Cloud's Vertex AI platform uses a class of Google-managed service accounts called Per-Project, Per-Product Service Agents, or P4SAs, to enable the Vertex AI service to act on resources within a customer's Google Cloud project. Each deployed AI agent built with Vertex AI's Agent Development Kit (ADK) and hosted on Agent Engine receives a P4SA at deployment time. The identity format follows the pattern `service-<PROJECT-ID>@gcp-sa-aiplatform-re.iam.gserviceaccount.com`. This P4SA is not provisioned by the deploying organization; it is created and managed by Google as part of the platform's operational model. The critical detail is that this service account is granted default permissions that exceed what any individual agent requires — including broad read access to Cloud Storage buckets and extensive OAuth 2.0 scopes that can extend to Google Workspace services.

This design reflects a common tension in managed cloud services: platform providers must grant their infrastructure accounts sufficient permissions to fulfill operational functions across a wide range of customer workloads, while customers deploying specific workloads need only a narrow subset of those permissions. For traditional managed services — databases, compute, networking — this tension is largely invisible because the platform account's actions are invisible to customer code. For AI agents, the dynamic is fundamentally different: agent code executes within the platform's infrastructure boundary and can interact with that infrastructure's credential surface. The metadata service — a standard Google Cloud mechanism that provides runtime credentials to any code running on GCP compute — is accessible from within the agent execution context and exposes the P4SA's credentials to the agent itself.

The history of this vulnerability class in Vertex AI predates the March 2026 disclosure. In November 2024, Unit 42 published research dubbed "ModeLeak" that identified two distinct privilege escalation paths within earlier Vertex AI components: one through the AI Platform Custom Code Service Agent, which held permissions to list all service accounts and create, delete, read, and write all storage buckets, and another through a model exfiltration vector where a malicious model deployment could access all fine-tuned models stored in the project's Cloud Storage [2]. Google addressed those specific issues, but the underlying architectural pattern — platform-provisioned service agents with permissions that exceed individual workload needs — persisted into the Agent Engine product generation. The March 2026 disclosure represents the same structural problem surfacing in a new and more operationally consequential context: AI agents with natural language interfaces and the ability to take autonomous actions.

Security Analysis

The Attack Chain

The attack path disclosed by Unit 42 researcher Ofir Shaty requires no vulnerability in the traditional sense. There is no memory corruption, no injection flaw, no authentication bypass. The entire chain relies on a design property: the P4SA's credentials are available to any code running within the agent's execution context via the GCP metadata service, and the P4SA holds more permissions than any single agent requires.

The sequence proceeds as follows. An attacker deploys an AI agent to Vertex AI Agent Engine — either by compromising a legitimate agent deployment or by deploying their own — using the ADK framework. When the agent is invoked, any call that triggers an outbound request from the agent runtime results in the agent's execution context querying the GCP metadata service. This is standard GCP behavior: the metadata service returns the credentials of the service account associated with the running compute resource. In the Agent Engine context, those credentials belong to the P4SA. The metadata service response exposes not only the credential material itself but also the GCP project identifier, the service agent's identity, and the OAuth 2.0 scopes granted to the machine hosting the agent runtime. None of this requires exploitation; it is information the metadata service is designed to provide to code running on the platform.

With those credentials in hand, an attacker pivots from the agent execution context into the customer's broader project environment. In Unit 42's proof-of-concept, the harvested P4SA credentials provided unrestricted read access to all Cloud Storage buckets within the project — encompassing any sensitive data the customer organization had stored there, including application data, model artifacts, configuration files, and deployment secrets. The same credentials exposed access to restricted Artifact Registry repositories, from which researchers retrieved private container images including Vertex AI Reasoning Engine core components and associated Dockerfiles, serialized code objects, and dependency manifests. The exposure extended bidirectionally: from the customer's data outward, and into Google's own infrastructure, where researcher access to private Artifact Registry repositories revealed internal software supply chain details [3][4][5].

Shaty's "double agent" framing captures the essential behavioral property of this attack: a compromised AI agent retains its outward functional behavior while conducting parallel malicious activity at the infrastructure layer. The agent continues answering queries, executing workflows, interacting with users — while simultaneously using its platform-granted credentials to conduct exfiltration or reconnaissance in the background. The victim organization has no reason to observe anomalous behavior at the agent's interface layer, because the interface layer is functioning normally. The malicious activity is occurring at

the infrastructure layer, using credentials the organization never explicitly provisioned and may not know exist. This detection challenge is particularly acute for AI agents, whose high-frequency, multi-system access patterns are difficult to distinguish from malicious background activity using conventional anomaly detection baselines.

The Vulnerability Class

The significance of the Vertex AI P4SA disclosure lies not in the specific permissions involved but in the structural pattern it demonstrates. Any cloud-hosted AI agent execution platform that provisions a platform-managed service account for agent workloads, grants that account permissions calibrated to the platform's operational breadth rather than individual agent requirements, and runs those agents within an execution environment where the metadata service is accessible faces a version of this risk.

The conditions required to trigger this vulnerability class are present across the cloud AI platform landscape. Managed agent execution environments — the category that includes Vertex AI Agent Engine, AWS Bedrock Agents, and analogous Azure services — provision infrastructure-level identities for their agent compute environments. The specific permissions, scoping mechanisms, and metadata service architectures differ by provider, but the underlying dynamic is consistent: a platform identity exists, it holds more permission than any individual agent requires, and the agent's execution context can interact with it. The structural conditions enabling this vulnerability class — platform-provisioned execution identities with permissions exceeding individual agent requirements — are present wherever AI platforms provision default service accounts for agent workloads, making this a cross-platform architectural risk, not a Google-specific product defect.

The threat model is further complicated by the multi-tenant nature of these platforms. When organizations deploy multiple AI agents across a shared cloud project, those agents may share a common P4SA. A compromise of one agent — through prompt injection, a malicious tool payload, a dependency supply chain compromise, or attacker-controlled input — can expose the credentials used by all agents in the project. The blast radius of a single agent compromise extends to the entire project's storage, model artifacts, and potentially beyond, depending on the permissions the P4SA holds.

Aggregate Risk Context

HiddenLayer's finding that 31% of organizations cannot determine whether they have experienced an AI security breach in the past year [1] suggests that enterprise AI agent deployments are scaling faster than security review processes can reliably track — precisely the conditions under which the P4SA vulnerability class poses maximum risk. HiddenLayer's 2026 AI Threat Landscape Report found that one in eight AI security breaches is now linked to agentic systems, reflecting the growing operational footprint of agents

in environments where security frameworks are still catching up [1]. The same report found that 76% of organizations now cite shadow AI as a definite or probable problem, up from 61% in 2025 [1]. These statistics describe an environment where AI agents are being deployed into production with insufficient visibility into their identity posture, permission boundaries, and runtime behavior — precisely the conditions under which the P4SA vulnerability class poses maximum risk.

Recommendations

Immediate Actions

Organizations currently operating Vertex AI agents using Agent Engine should treat default P4SA permissions as a misconfiguration requiring remediation rather than an acceptable default. The primary immediate action is adoption of the BYOSA (Bring Your Own Service Account) architecture that Google now explicitly recommends in updated documentation. Under BYOSA, the default P4SA is replaced by an organization-controlled service account provisioned with only the permissions required by the specific agent's functional scope — typically a narrow subset of the default. This single change removes the platform-provisioned over-privilege that enables the attack chain described above.

As part of P4SA remediation, organizations should audit the OAuth 2.0 scopes associated with their agent execution environments. Scopes that extend to Google Workspace services — Gmail, Calendar, Drive — are almost never required for an AI agent's operational function and represent unnecessary exposure if the execution context is compromised. Restricting scopes to the minimum required for the agent's specific tool integrations and data access patterns eliminates this exposure independently of storage permission changes.

Organizations should also immediately inventory any existing Vertex AI deployments to determine which are using default P4SA credentials versus organization-supplied service accounts. For deployments in production using default P4SAs, the BYOSA migration should be treated with the urgency of a security remediation rather than scheduled as a future improvement.

Short-Term Mitigations

Beyond addressing Vertex AI specifically, organizations should extend their service account review to all cloud-hosted AI agent execution environments, including AWS Bedrock Agents, Azure AI Agent Service, and any third-party managed agent platforms operating within their cloud environments. The review should examine, for each platform, what identity is used for agent execution, what permissions that

identity holds, whether the metadata service or equivalent credential endpoint is accessible from within the agent execution context, and whether the platform provides a mechanism to substitute a least-privilege organization-controlled identity.

Monitoring and detection capabilities for AI service account activity represent a gap in most organizations' current tooling. Conventional SIEM and CSPM solutions alert on anomalous IAM activity based on patterns established by human-operated workloads; the access patterns of AI agents — high-frequency, multi-system, tool-driven — may not be well-distinguished from anomalous human activity by existing baselines. Organizations should establish explicit behavioral baselines for AI agent service account activity, including expected storage access patterns, API call volumes per hour, and time-of-day profiles, and configure alerts for deviations that may indicate credential harvest and misuse.

Prior to deploying any new AI agent into production, organizations should require a documented permission validation step confirming that the service account used for execution holds only the permissions demonstrably required by the agent's function. This validation should be a formal gate in the agent deployment pipeline — the AI equivalent of a firewall rule review before network change deployment — not an optional post-deployment check.

Strategic Considerations

The P4SA vulnerability class is a foreseeable consequence of deploying autonomous systems into infrastructure environments designed around the assumption that workloads are human-authored, human-reviewed, and human-monitored. AI agents are none of those things at runtime. They execute autonomously, often at machine speed, with access to credentials and tool integrations that a human engineer would recognize as sensitive but that agents operating without explicit credential-handling guardrails have no built-in capacity to recognize or treat with care. Addressing this structurally requires extending the principle of least privilege — a well-established cloud security discipline — into the AI agent lifecycle as a non-negotiable deployment requirement, not a best practice.

Cloud AI platform providers bear a design responsibility in this regard. Provisioning over-privileged default service accounts for agent execution environments creates a latent vulnerability in every customer deployment that defaults to platform behavior. The BYOSA recommendation that Google has issued following the Unit 42 disclosure is sound, but it places the remediation burden on customers who may not be aware of the risk or may lack the IAM expertise to configure least-privilege service accounts correctly. A more defensible default would provision a narrowly scoped service account and require explicit customer action to expand permissions — inverting the current posture where over-privilege is the default and least privilege requires additional configuration.

For organizations with significant AI agent footprints, this incident should prompt a broader review of non-human identity governance. AI agents, automation scripts, CI/CD pipelines, and data processing workloads collectively constitute a non-human identity population that typically receives less rigorous lifecycle governance than human identities: service accounts are created for specific tasks and forgotten, permissions are granted permissively to unblock deployment and never revisited, and credential rotation may be inconsistent. A structured non-human identity inventory — covering creation, permission review, rotation schedule, and decommissioning criteria — that explicitly includes AI agent service accounts is a prerequisite for managing this risk class at scale.

CSA Resource Alignment

The Vertex AI P4SA vulnerability class maps directly to several layers of the CSA MAESTRO framework for AI threat modeling. MAESTRO Layer 3 (Infrastructure & Scalability) addresses the cloud infrastructure execution environment where AI agents operate, including the identity and access management configuration of the compute and storage resources that support agent workloads. The over-privilege finding is a Layer 3 misconfiguration: the infrastructure identity associated with the agent execution environment holds permissions that exceed the agent's operational requirements, creating a lateral movement path into broader project resources. MAESTRO Layer 4 (Agent Interaction & Orchestration) addresses how agents interact with tools, external services, and each other; the metadata service credential harvest that enables this attack occurs at the Layer 4/3 boundary, where the agent's execution context intersects with the underlying infrastructure.

The CSA AI Controls Matrix (AICM) provides the control framework most directly applicable to remediating this risk. AICM controls in the Identity and Access Management domain address non-human identities, service account governance, and the principle of least privilege in cloud environments. Because AICM is a superset of the Cloud Controls Matrix (CCM), CCM IAM controls — particularly those in the Identity and Access Management domain addressing service accounts and role assignment — also apply. Organizations using CCM for compliance assessments should recognize that AI agent service accounts fall within the scope of existing IAM controls and should be assessed accordingly, not treated as a separate category exempt from standard governance.

The BYOSA remediation recommended by Google aligns closely with CSA's Zero Trust guidance, which emphasizes that all workloads — human and automated — should receive only the permissions required for their specific function, verified continuously, without inheriting ambient platform-level trust. The Zero Trust model applied to AI agents means that each agent receives an identity scoped to its individual

function, its access to resources is authorized based on that identity, and its behavior at runtime is monitored against expected patterns. This is the operational model that BYOSA enables when combined with behavioral monitoring and explicit permission boundary documentation.

Organizations pursuing STAR (Security Trust Assurance and Risk) certification for AI system deployments should treat AI agent service account configuration as a STAR audit criterion. The question of whether deployed agents use default platform-provisioned credentials or organization-controlled least-privilege service accounts is directly assessable and should be included in AI security attestations submitted to customers and auditors. As CSA's STAR program develops AI-specific assurance criteria, the P4SA vulnerability class — and the BYOSA control — represent a concrete, verifiable security practice that STAR assessments can require.

References

[1] HiddenLayer. "HiddenLayer Releases the 2026 AI Threat Landscape Report." HiddenLayer, March 18, 2026.

[2] Palo Alto Networks Unit 42. "ModeLeak: Privilege Escalation to LLM Model Exfiltration in Vertex AI." Palo Alto Networks Unit 42 Blog, November 12, 2024.

[3] Palo Alto Networks Unit 42 (Ofir Shaty). "Double Agents: Exposing Security Blind Spots in GCP Vertex AI." Palo Alto Networks Unit 42 Blog, March 31, 2026.

[4] The Hacker News. "Vertex AI Vulnerability Exposes Google Cloud Data and Private Artifacts." The Hacker News, March 31, 2026.

[5] CyberSecurityNews. "Google Cloud's Vertex AI Platform Vulnerability Allows Attackers to Access Sensitive Data." CyberSecurityNews, April 1, 2026.