INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-07-02 14:42 UTC

Structural Identity Gap in Enterprise AI Agent Deployments Undermines Access Control and Auditability

GOVERNANCE | HIGH | CVSS 7.5

SCC Item ID	SCC-GOV-2026-0090
Type	Governance
Severity	HIGH
CVSS Base Score	7.5
Affected Products	OAuth 2.1 / JWT (RFC 9068), Model Context Protocol (MCP), Claude Code, GitHub, general applicability across enterprise AI agent deployments
Discovery Source	Rss:T1 Threatintel

Executive Summary

Enterprise AI agents, automated systems that take actions on behalf of users, lack standardized mechanisms to declare who they are, whose authority they act under, and what they are permitted to do. Current identity standards (OAuth 2.1, JWT) carry no fields for agent instance identity or human-to-agent delegation, meaning organizations cannot enforce least-privilege access controls on AI agents or produce audit trails that satisfy compliance requirements. According to CrowdStrike and Aembit technical analysis, this structural gap exposes enterprises to unauthorized data access, privilege abuse, and undetectable out-of-scope agent behavior across any deployment using the Model Context Protocol or similar agent-to-tool integration frameworks.

Technical Analysis

The identity gap manifests at the token layer: OAuth 2.1 access tokens and JWTs conforming to RFC 9068 contain no standardized claims for agent instance identity, the human principal delegating authority, the scope of that delegation, or the session binding between agent and principal. The Model Context Protocol (MCP), which defines how AI agents invoke tools and external services, currently provides no mechanism to carry or enforce this context. As a result, an AI agent operating under a user's OAuth token is indistinguishable from the user at the authorization layer, making fine-grained access control and meaningful audit logging structurally impossible without custom instrumentation. CVSS scoring does not apply to structural standards gaps; severity is rated qualitatively based on compliance urgency and exposure scope. Relevant CWEs: CWE-284 (Improper Access Control), CWE-287 (Improper Authentication), CWE-269 (Improper Privilege Management), CWE-522

(Insufficiently Protected Credentials), CWE-732 (Incorrect Permission Assignment for Critical Resource). MITRE ATT&CK techniques of concern include T1078 (Valid Accounts), T1528 (Steal Application Access Token), T1550 (Use Alternate Authentication Material), T1134 (Access Token Manipulation), T1548 (Abuse Elevation Control Mechanism), T1199 (Trusted Relationship), and T1552.001 (Credentials In Files). No CVE is assigned; this is a structural design gap in current standards, not a discrete vulnerability in a single product. CrowdStrike has announced a continuous identity capability for AI agents. Aembit has published technical analysis covering MCP, OAuth 2.1, and PKCE authorization gaps. Industry analysis suggests organizations with AI-expanded identity footprints experience higher breach rates; direct verification from primary research sources is recommended before citing specific statistics.

Action Checklist

1. Step 1: Containment. Audit all AI agent deployments that operate under delegated user credentials (OAuth tokens, service account JWTs). Identify which agents have access to production systems, sensitive data stores, or external services via MCP or similar tool-invocation protocols. Restrict agent OAuth scopes to the minimum required using NIST AC-6 (Least Privilege) and CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts) as the governing standard. Where scope reduction is not immediately feasible, isolate agent workloads from sensitive data paths.
2. Step 2: Detection. Enable and review logs for all OAuth token issuance and usage events associated with AI agent service accounts or client IDs. Query for tokens with broad scopes (e.g., read:all, write:all, admin) issued to non-human principals. Flag any access events where the acting principal cannot be traced to a human initiator in the audit record. Apply NIST AU-2 (Event Logging) and AU-3 (Content of Audit Records) to verify that agent-originated actions are distinguishable from human-originated actions in your SIEM. Use NIST SI-4 (Information System Monitoring) to detect anomalous lateral movement by agent identities.
3. Step 3: Eradication. Implement custom JWT claims or a proprietary token enrichment layer to carry agent instance ID, delegating human principal, and delegation scope until standards bodies (IETF, MCP governance) ratify standardized fields. For MCP-connected agents specifically, review Aembit's published technical analysis on PKCE and authorization gaps and apply PKCE enforcement where supported. Rotate all service account credentials used by AI agents and scope them narrowly per CIS 5.2 (Use Unique Passwords) and NIST IA-5 (Authentication and Authorization).
4. Step 4: Recovery. Validate that agent-originated access events now carry sufficient identity context to satisfy audit requirements under NIST AU-3 and AU-10 (Non-Repudiation). Test that access control enforcement matches the intended delegation scope for each deployed agent. Implement NIST AC-2 (Account Management) review cycles to verify agent permissions have not drifted from their defined scope. Confirm that your SIEM can generate agent-specific audit reports distinct from human user reports.
5. Step 5: Post-Incident. Document the identity architecture for every AI agent in production, including the human principal, delegation chain, token scope, and connected tools or services. Map gaps against NIST AC-3 (Access Enforcement), AC-5 (Separation of Duties), and CIS 6.1 (Establish an Access Granting Process). Establish a recurring review cadence for agent identity posture as MCP and OAuth standards evolve. Engage your GRC function to assess whether current agent audit trails satisfy regulatory audit requirements applicable to your industry.

IR / Forensic Enrichment

Triage Priority	URGENT
Escalation Criteria	Escalate immediately to CISO and GRC if any AI agent operating under delegated OAuth credentials is found to have accessed PII, PHI, financial records, or systems in scope for SOC 2, HIPAA, PCI-DSS, or SEC regulations, as the absence of human-attributable audit records for those access events may constitute a reportable compliance failure or breach notification trigger under applicable law.
Recovery Notes	Before returning any AI agent to production, require a signed-off validation checklist confirming: custom delegation claims are present in all issued tokens, MCP tool-invocation sessions log the human principal chain, and the authorization server enforces the narrowed scope without exception. Monitor agent OAuth token issuance and usage logs daily for the first 30 days post-recovery, specifically watching for scope creep (tokens issued with scopes broader than the post-remediation baseline) and for any access events still lacking `delegated_by` attribution. Continue monitoring on a weekly cadence thereafter until IETF or MCP governance ratifies standardized agent identity fields and your token infrastructure is updated to implement them natively.
Forensic Artifacts	Authorization server token issuance logs: entries where `client_type` is non-human (service account, agent client ID) and `scope` contains broad grants (admin, write:all, read:all) without a corresponding human-initiator claim — the structural absence of a delegation field is the primary indicator of this governance gap. | MCP tool-invocation session logs: records of external service calls (GitHub API, data stores, third-party SaaS) initiated by an agent client ID where no parent human session ID or delegation chain can be reconstructed — these represent the unattributable access events that are the core risk. | JWT payload archives (base64-decoded): pre-remediation snapshots of agent-issued tokens showing the absence of `agent_instance_id`, `act`, or `on_behalf_of` claims in the RFC 9068 JWT structure, preserved as evidence of the identity gap state at time of discovery. | Cloud IAM / API gateway access logs: CloudTrail `AssumeRoleWithWebIdentity` events, Azure AD sign-in logs, or GCP Cloud Audit Logs filtered to non-human principals, showing the resource access footprint of each AI agent (Claude Code, MCP-connected agents) prior to scope restriction. | GitHub audit log export (for Claude Code or GitHub-integrated agents): `GET /orgs/{org}/audit-log` filtered to `oauth_access` and `repo.access` actions by agent app client IDs, documenting which repositories and API endpoints were accessed under the unattributed agent identity.

Per-Action IR Details

Step 1: Containment — Audit all AI agent deployments that operate under delegated user credentials (OAuth tokens, service account JWTs). Identify which agents have access to production systems, sensitive data stores, or external services via MCP or similar tool-invocation protocols. Restrict agent OAuth scopes to the minimum required using NIST AC-6 (Least Privilege) and CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts) as the governing standard. Where scope reduction is not immediately feasible, isolate agent workloads from sensitive data paths.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment Strategy: prioritize isolating affected assets to prevent further exposure while preserving operational continuity where possible.

Controls: NIST AC-6 (Least Privilege), NIST AC-3 (Access Enforcement), 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 `az ad sp list --all --query '[].{DisplayName:displayName, AppId:appId}' -o table` (Azure) or `gcloud iam service-accounts list` (GCP) to enumerate all service accounts. Cross-reference with a manually maintained spreadsheet of known AI agent client IDs. Use `curl -H 'Authorization: Bearer ' https:///introspect` to inspect active token scopes. Block over-scoped agent client IDs at the API gateway level using nginx `deny` directives or iptables rules targeting agent egress IPs while full scope reduction is planned.

Evidence: Before restricting or revoking any OAuth token or JWT, capture: (1) a full dump of currently active tokens and their scopes from your authorization server (e.g., `SELECT client_id, scope, expires_at FROM oauth_tokens WHERE subject NOT IN (SELECT user_id FROM human_users)` against your auth-server DB); (2) MCP tool-invocation logs showing which external services each agent client ID contacted, including timestamps and resource paths; (3) a snapshot of current agent-to-resource access mappings from your cloud IAM policy engine. These are volatile — token revocation destroys the live scope state.

Step 2: Detection — Enable and review logs for all OAuth token issuance and usage events associated with AI agent service accounts or client IDs. Query for tokens with broad scopes (e.g., read:all, write:all, admin) issued to non-human principals. Flag any access events where the acting principal cannot be traced to a human initiator in the audit record. Apply AU-2 (Event Logging) and AU-3 (Content of Audit Records) to verify that agent-originated actions are distinguishable from human-originated actions in your SIEM. Use D3-LAM (Local Account Monitoring) to detect anomalous lateral movement by agent identities.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: correlate log sources to identify the scope of agent identity abuse and establish whether agent actions can be attributed to an authorizing human principal.

Controls: NIST AU-2 (Event Logging), NIST AU-3 (Content Of Audit Records), NIST AU-6 (Audit Record Review, Analysis, And Reporting), CIS 8.2 (IG1/IG2/IG3) — Collect Audit Logs

Compensating: Without a SIEM, use `jq` to parse authorization server JSON logs: `cat oauth_access.log | jq 'select(.client_type=="service_account" and (.scope | test("admin|write:all|read:all")))' > flagged_agent_tokens.json`. For GitHub-connected Claude Code agents, query the GitHub audit log API: `GET /orgs/{org}/audit-log?phrase=action:oauth_access`. To detect missing human-initiator attribution, grep application logs for JWT `sub` fields that match agent client ID patterns but lack a `act` (actor) or custom delegation claim: `grep -E '"sub":"agent-[a-z0-9]+"' app.log | grep -v '"delegated_by"'`. Use osquery `SELECT * FROM user_events WHERE username LIKE 'svc-agent%';` on hosts where agents execute.

Evidence: The structural identity gap means the primary forensic indicator is absence, not presence: look for OAuth access log entries where `sub` is a service account or agent client ID but no `act`, `on_behalf_of`, or custom delegation claim exists in the JWT payload. Capture: (1) raw authorization server token issuance logs (typically `/var/log/oauth2-server/token.log` or cloud-provider audit trails like AWS CloudTrail `AssumeRoleWithWebIdentity` events) before any log rotation; (2) MCP session logs showing tool invocation chains without a traceable human session ID; (3) GitHub Actions or Claude Code execution logs showing repository or API access under agent identity. These logs are volatile if log rotation is active — archive immediately.

Step 3: Eradication — Implement custom JWT claims or a proprietary token enrichment layer to carry agent instance ID, delegating human principal, and delegation scope until standards bodies (IETF, MCP governance) ratify standardized fields. For MCP-connected agents specifically, review Aembit's published technical analysis on PKCE and authorization gaps and apply PKCE enforcement where supported. Rotate all service account credentials used by AI agents and scope them narrowly per CIS 5.2 (Use Unique Passwords) and D3-CRO (Credential Rotation).

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication: remove the structural conditions enabling unattributable agent access by enforcing token enrichment and credential hygiene before restoring agent workloads.

Controls: NIST AC-6 (Least Privilege), NIST IA (Identification and Authentication) — note: IA-family controls govern credential management; cite IA as the applicable family for credential rotation procedures in the absence of a specific IA control ID in the verified knowledge base, CIS 5.2 (IG1/IG2/IG3) — Use Unique Passwords

Compensating: Use a lightweight JWT middleware (e.g., a Python Flask decorator or nginx `auth_request` module) to intercept agent token issuance and inject custom claims: `agent_instance_id`, `delegated_by` (human principal UPN), and `delegation_scope` (enumerated permitted actions). Store the enrichment mapping in a local SQLite registry. For PKCE enforcement on MCP OAuth flows where the authorization server supports it, set `require_pkce=true` in your OAuth server config (e.g., Keycloak: `pkceCodeChallengeMethod=S256`). Rotate service account credentials using: `openssl rand -base64 32` for new secrets and update via your IdP's API. Document each rotation in a timestamped change log.

Evidence: Before rotating any service account credential or reconfiguring token issuance, capture: (1) the current JWT payload of all active agent tokens (`echo | base64 -d | jq .`) to record the pre-rotation scope and claims state as a forensic baseline; (2) a network capture (`tcpdump -i any -w agent_mcp_traffic_prerotation.pcap port 443`) of any active MCP tool-invocation sessions, since PKCE enforcement will terminate non-compliant flows; (3) the current OAuth client configuration (client ID, secret hash, allowed scopes, redirect URIs) from your authorization server admin API before changes are applied. Credential rotation destroys the live secret state — baseline capture is mandatory.

Step 4: Recovery — Validate that agent-originated access events now carry sufficient identity context to satisfy audit requirements under NIST AU-3 and AU-10 (Non-Repudiation). Test that access control enforcement matches the intended delegation scope for each deployed agent. Implement D3-UAP (User Account Permissions) review cycles to verify agent permissions have not drifted from their defined scope. Confirm that your SIEM can generate agent-specific audit reports distinct from human user reports.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery: restore AI agent workloads to operation only after verifying that token enrichment, access controls, and audit logging meet the identity attribution requirements that were absent prior to this remediation.

Controls: NIST AU-3 (Content Of Audit Records), NIST AU-10 (Non-Repudiation), NIST AU-6 (Audit Record Review, Analysis, And Reporting), CIS 6.1 (IG1/IG2/IG3) — Establish an Access Granting Process

Compensating: Without a SIEM, write a Python or bash validation script that samples the last 100 agent-issued JWT access events from your authorization server log and confirms presence of `agent_instance_id` and `delegated_by` claims in every record: `jq 'select(.sub | test("agent-")) | {sub, agent_instance_id, delegated_by, scope}' oauth_access.log | grep -c '"delegated_by": null'` — a non-zero result indicates incomplete enrichment. Generate agent-specific access reports by filtering on agent client ID patterns in raw logs using `awk` or `grep -E`. Use a spreadsheet to manually track permission drift against the post-eradication scope baseline documented in Step 3.

Evidence: At this phase, evidence capture shifts from volatile forensics to validation artifacts: (1) generate a test JWT from each recovered agent client ID and decode the payload to confirm all required custom claims are present; (2) issue a controlled test access request from each agent against a resource at the boundary of its delegation scope and verify the authorization server enforces the scope limit (confirm deny logs appear in AU-2 event log); (3) archive the post-recovery token schema and access policy configuration as the new compliance baseline for future drift detection. These are configuration-state artifacts — no live-state destruction risk at this phase, but capture before the next configuration change cycle.

Step 5: Post-Incident — Document the identity architecture for every AI agent in production, including the human principal, delegation chain, token scope, and connected tools or services. Map gaps against NIST AC-3 (Access Enforcement), AC-5 (Separation of Duties), and CIS 6.1 (Establish an Access Granting Process). Establish a recurring review cadence for agent identity posture as MCP and OAuth standards evolve. Engage your GRC function to assess whether current agent audit trails satisfy regulatory audit requirements applicable to your industry.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: conduct lessons-learned analysis, update IR playbooks to address AI agent identity as an explicit threat surface, and share findings to improve future detection of structural identity gaps in enterprise AI deployments.

Controls: NIST AC-3 (Access Enforcement), NIST AC-5 (Separation Of Duties), NIST AU-11 (Audit Record Retention), CIS 6.1 (IG1/IG2/IG3) — Establish an Access Granting Process, CIS 3.2 (IG1/IG2/IG3) — Establish and

Maintain a Data Inventory

Compensating: Maintain an AI Agent Identity Register as a version-controlled Markdown or CSV file in your internal wiki or Git repository, with columns: agent_name, client_id, human_owner, delegation_scope, connected_tools (MCP endpoints, GitHub repos, APIs), last_reviewed, and compliance_notes. Set a calendar reminder for quarterly review aligned to IETF OAuth working group and MCP governance release cycles. For GRC gap mapping, use the NIST SP 800-53 Rev. 5 control catalog (publicly available at csrc.nist.gov) as a checklist against each agent's documented identity architecture — no paid tooling required.

Evidence: Post-incident documentation artifacts to preserve as permanent record: (1) the pre-remediation token scope inventory captured in Step 1 — this establishes the gap baseline for regulatory or audit purposes; (2) the custom claims schema implemented in Step 3, including the token enrichment middleware configuration and the delegation scope registry, as evidence of control implementation; (3) AU-3-compliant sample audit records from Step 4 validation demonstrating that agent-originated actions are now attributable to a human principal — these serve as compliance evidence if regulators (e.g., SEC, HIPAA OCR, SOC 2 auditors) request proof of AI agent access auditability. Retain per your AU-11 (Audit Record Retention) policy period.

Detection Guidance

There is no discrete exploit signature to detect; the risk is structural. Detection focuses on identifying AI agents operating with over-broad credentials and audit trails that cannot distinguish agent actions from human actions. Query your identity provider logs for OAuth tokens issued to client IDs associated with AI agent frameworks (e.g., Claude Code, GitHub Copilot agent mode, MCP-connected tools). Flag any token with scopes broader than the agent's documented function. In your SIEM, search for access events where the user principal matches a service account or non-interactive account pattern but the action pattern (high-volume API calls, sequential tool invocations, off-hours data access) is inconsistent with human behavior. Apply NIST AU-6 (Audit Record Review, Analysis, and Reporting) review cadence to agent-associated accounts. Use NIST SI-4 (Information System Monitoring) to surface anomalous access patterns from agent identities. Absence of an agent instance identifier or delegating human principal in access log records is itself a structural indicator of the gap.

Framework Mappings

MITRE-ATTACK

- **T1078** — Valid Accounts
- **T1528** — Steal Application Access Token
- **T1550** — Use Alternate Authentication Material
- **T1059** — Command and Scripting Interpreter
- **T1552.001** — Credentials In Files
- **T1134** — Access Token Manipulation
- **T1199** — Trusted Relationship
- **T1550.001** — Application Access Token
- **T1548** — Abuse Elevation Control Mechanism

NIST-800-53R5

- **AC-2** — Account Management
- **AC-6** — Least Privilege

- **IA-2** — Identification and Authentication (Organizational Users)
- **IA-5** — Authenticator Management
- **CM-7** — Least Functionality
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **SI-7** — Software, Firmware, and Information Integrity
- **CM-6** — Configuration Settings
- **IA-8** — Identification and Authentication (Non-Organizational Users)
- **AC-3** — Access Enforcement

OWASP-TOP10-2021

- **A01:2021** — Broken Access Control
- **A07:2021** — Identification and Authentication Failures
- **A04:2021** — Insecure Design

CIS-V8

- **5.4** — Restrict Administrator Privileges to Dedicated Administrator Accounts
- **6.8** — Define and Maintain Role-Based Access Control
- **6.3** — Require MFA for Externally-Exposed Applications
- **6.4** — Require MFA for Remote Network Access
- **6.5** — Require MFA for Administrative Access
- **3.3** — Configure Data Access Control Lists
- **5.2** — Use Unique Passwords
- **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(d)** — Person or Entity Authentication
- **164.308(a)(5)(ii)(D)** — Password Management
- **164.312(a)(1)** — Access Control

NIST-CSF-2

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

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1078	Valid Accounts	Defense-Evasion
T1528	Steal Application Access Token	Credential-Access
T1550	Use Alternate Authentication Material	Defense-Evasion
T1059	Command and Scripting Interpreter	Execution
T1552.001	Credentials In Files	Credential-Access
T1134	Access Token Manipulation	Defense-Evasion
T1199	Trusted Relationship	Initial-Access
T1550.001	Application Access Token	Defense-Evasion
T1548	Abuse Elevation Control Mechanism	Privilege-Escalation

Sources

Source	URL	Tier
Blog	https://www.crowdstrike.com/en-us/blog/the-identity-problem-hiding-...	T1
CrowdStrike	https://www.crowdstrike.com/en-us/blog/crowdstrike-announces-contin...	T1
Cybersecurity Insiders	https://www.cybersecurity-insiders.com/your-ai-assistant-might-be-y...	T3
Recordedfuture	https://www.recordedfuture.com/research/emerging-enterprise-securit...	T1
MCP, OAuth 2.1, PKCE, and the Future of AI Authorization - Aembit	https://aembit.io/blog/mcp-oauth-2-1-pkce-and-the-future-of-ai-auth...	T3

Generated 2026-07-02 14:42 UTC by TJS Security Command Center