INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-06-28 15:15 UTC

AI Agent Identity Has No Standard: Why OAuth Tokens Are Blind to Agentic Context

GOVERNANCE | HIGH | CVSS 7.5

SCC Item ID	SCC-GOV-2026-0087
Type	Governance
Severity	HIGH
CVSS Base Score	7.5
Affected Products	OAuth 2.1 implementations, JWT-based identity systems (RFC 9068), AI agent frameworks including Claude Code and MCP-based agents, CrowdStrike Falcon Identity Security
Discovery Source	Rss:T1 Threatintel

Executive Summary

Enterprise AI agents operating under OAuth 2.1 and JWT-based identity systems carry no standardized way to distinguish agent actions from human actions, assign agent-specific identity, or enforce least-privilege boundaries for agentic workflows. Every organization deploying AI agents across HR, software development, or customer-facing operations is affected, because the gap exists in the underlying identity standards, not in a single patchable product. The business risk is material: a compromised agent credential grants broad, persistent access with no audit trail linking actions to an agent instance, creating uncontrolled privilege abuse potential and significant accountability exposure.

Technical Analysis

OAuth 2.1 and JWT-formatted access tokens (RFC 9068) contain no standardized claims for agent instance identity, delegating-user context, or the agent-user trust relationship. When AI agents authenticate using these tokens, the identity layer cannot distinguish agent-initiated requests from human-initiated requests, cannot scope permissions to agentic context, and cannot generate audit records that satisfy least-privilege or non-repudiation requirements. Affected systems include any deployment of Claude Code, MCP-based agents, CrowdStrike Falcon Identity Security integrations, or custom agent frameworks relying on standard OAuth 2.1 flows. No CVE is assigned; this is a structural standards gap. No CVSS score applies; severity is qualitative, based on compliance urgency and exposure scope. Relevant CWE mappings: CWE-287 (Improper Authentication), CWE-269 (Improper Privilege Management), CWE-1270 (Generation of Incorrect Security Tokens), CWE-284 (Improper Access Control). MITRE ATT&CK techniques in scope: T1550.001 (Application Access Token), T1134 (Access Token Manipulation), T1098 (Account Manipulation), T1078 (Valid Accounts), T1078.004 (Cloud Accounts), T1548 (Abuse Elevation Control Mechanism). No patch exists at the standards

level; mitigations are vendor-specific and compensating in nature. CrowdStrike has published guidance on continuous identity security for AI agents. The MCP community is developing authentication and authorization guidance, but no ratified cross-industry standard exists as of 2026-03-04.

Action Checklist

1. Step 1: Assessment, Audit all active OAuth tokens and service accounts used by AI agents across HR, dev tooling, and customer-facing workflows. Revoke any tokens with scopes broader than the agent's documented minimum function. For MCP-based and Claude Code deployments, restrict token lifetimes and enforce short expiry windows to limit blast radius of a compromised agent credential. Reference: NIST AC-6 (Least Privilege), CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts).
2. Step 2: Detection, Enable and review identity provider logs for OAuth token issuance and usage. Query for tokens issued to non-human service principals that accessed sensitive scopes (e.g., HR data APIs, code repositories, customer record systems). Look for token reuse across multiple source IPs or user-agent strings inconsistent with your agent framework's expected behavior. Correlate against AU-2 (Event Logging) requirements: log what type of action occurred, when, where, and under which identity. Reference: NIST AU-2, NIST AU-12, CIS 8.2 (Collect Audit Logs).
3. Step 3: Implementation, Implement compensating controls at the application layer. Add custom JWT claims or metadata headers to agent requests to carry agent instance ID, originating user context, and agent-user relationship. Enforce these claims in authorization middleware so any request lacking agent context is rejected. Where possible, issue agent-specific service accounts with scopes limited to named resources, not broad API permissions. Reference: NIST AC-3 (Access Enforcement), NIST AC-6 (Least Privilege).
4. Step 4: Validation, Validate that revised token issuance policies are producing auditable records distinguishing agent actions from human actions. Test that short-lived agent tokens expire and rotate as configured. Confirm that revoked legacy tokens are no longer accepted by downstream services. Monitor identity provider logs for anomalous token usage patterns for 30 days post-remediation. Reference: NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST AU-9 (Protection of Audit Information).
5. Step 5: Governance, Document the agent identity gap as a standing control deficiency in your risk register. Assign ownership to track the MCP authentication specification and any emerging IETF or NIST guidance on agentic identity. Establish a policy requiring all new AI agent deployments to include a documented identity model, minimum-scope token definition, and audit logging plan before production authorization. Reference: NIST AC-1 (Policy and Procedures), NIST AC-5 (Separation of Duties), CIS 7.1 (Establish and Maintain a Vulnerability Management Process).

IR / Forensic Enrichment

Triage Priority	URGENT

Escalation Criteria	Escalate immediately if IdP logs reveal any agent service account token used from a source IP outside the agent runtime's known egress range, if any agent token is found to have accessed HR PII endpoints or customer record systems beyond its documented scope (potential GDPR/HIPAA breach notification trigger), or if the organization lacks any mechanism to distinguish agent-initiated from human-initiated actions in its audit logs — indicating the compensating controls in Steps 2 and 3 cannot be implemented without external identity engineering support.
Recovery Notes	Post-containment, maintain a 30-day enhanced monitoring window on all IdP token issuance events for agent service principals, specifically alerting on any token grant that includes scopes broader than the minimum defined in the remediated service account definitions. Verify that every AI agent deployment — including Claude Code sessions and MCP-based integrations — is producing audit records with agent-context claims present before marking the incident closed; absence of `agent_id` or equivalent claim in post-remediation logs means the eradication step did not fully propagate to that deployment. The structural identity gap in OAuth 2.1 and RFC 9068 is not patchable at the product level and will persist until IETF or NIST issues normative guidance on agentic identity, so the risk register entry from Step 5 must remain open and actively tracked.
Forensic Artifacts	IdP OAuth token grant logs (Okta System Log event type `app.oauth2.token.grant.*` or Azure AD Audit Log category `ApplicationManagement`) for all service principals with 'agent', 'bot', 'mcp', or 'claude' in the display name — the absence of agent-context claims in issued JWTs is the primary forensic indicator of this control gap | Decoded JWT payloads from captured agent requests to HR data APIs, code repository APIs, and CRM endpoints — specifically the `scope`, `sub`, `aud`, and custom claims fields; a JWT with broad scopes and no `agent_id`, `on_behalf_of`, or `delegation_chain` claim confirms exploitation surface | MCP server session state and tool invocation logs (typically at `~/.mcp/sessions/` or the configured MCP working directory) showing which tools were invoked, under which identity, and whether any human-delegated context was recorded at the time of invocation | API gateway or reverse-proxy access logs filtered for requests where `Authorization: Bearer` is present and the `User-Agent` string does not match the expected agent framework identifier — mismatched User-Agent on a service account token is a strong indicator of credential reuse or token theft outside the agent runtime | Service account credential issuance history from the IdP showing token lifetime settings, refresh token grants, and scope assignments at the time of each agent deployment — documents whether minimum-scope and short-expiry policies were in place before remediation, establishing the pre-remediation risk baseline for audit purposes

Per-Action IR Details

Step 1: Containment — Audit all active OAuth tokens and service accounts used by AI agents across HR, dev tooling, and customer-facing workflows. Revoke any tokens with scopes broader than the agent's documented minimum function. For MCP-based and Claude Code deployments, restrict token lifetimes and enforce short expiry windows to limit blast radius of a compromised agent credential. Reference: NIST AC-6 (Least Privilege), CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts).

NIST Phase: Containment

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

Controls: NIST AC-6 (Least Privilege), NIST AC-12 (Session Termination), CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts)

Compensating: Use your IdP's native CLI to enumerate and revoke tokens at scale. For Azure AD/Entra: `az ad app list --query "[].{appId:appId, displayName:displayName}"` then `az ad app credential list --id ` to surface agent service principals. For Okta: query the `/api/v1/apps` endpoint with `curl` and the Okta API token to list all service app

assignments and active token grants. Revoke via `/api/v1/apps/{appId}/grants` DELETE calls. Log every revocation with timestamp and scope removed to a local CSV for audit trail. Two-person team: one queries and triages scope width, one executes revocations and records them.

Evidence: BEFORE revoking any token, capture a full export of active OAuth token grants including: token subject (`sub`), client_id, granted scopes, issued-at (`iat`), expiry (`exp`), and last-used timestamp from your IdP's token introspection endpoint (RFC 7662: `POST /introspect`). For MCP-based agents, capture the agent's current process list and any open network connections (`netstat -ano` on Windows or `ss -tulnp` on Linux) to document which agent processes hold live token state. This establishes the pre-revocation blast-radius baseline and is destroyed once tokens are invalidated. Preserve the raw IdP token grant export as an immutable artifact (hash with SHA-256 before any remediation action).

Step 2: Detection — Enable and review identity provider logs for OAuth token issuance and usage. Query for tokens issued to non-human service principals that accessed sensitive scopes (e.g., HR data APIs, code repositories, customer record systems). Look for token reuse across multiple source IPs or user-agent strings inconsistent with your agent framework's expected behavior. Correlate against AU-2 (Event Logging) requirements: log what type of action occurred, when, where, and under which identity. Reference: NIST AU-2, NIST AU-12, CIS 8.2 (Collect Audit Logs), D3-LAM (Local Account Monitoring).

NIST Phase: Detection Analysis

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

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

Compensating: Without a SIEM, use IdP-native log exports and jq for analysis. Export Okta System Log or Azure AD Sign-In Logs as JSON, then use `jq '[.[] | select(.actor.type == "ServicePrincipal" or .actor.type == "SystemPrincipal")]'` to isolate non-human principals. For JWT inspection without tooling, pipe base64-decoded JWT payloads through `python3 -c "import sys,base64,json; p=sys.argv[1]+'=='; print(json.dumps(json.loads(base64.urlsafe_b64decode(p)), indent=2))"` to verify whether agent-issued tokens carry any agent-context claims (`agent_id`, `on_behalf_of`, `delegation_chain`). Flag tokens where `sub` is a service account but `User-Agent` in access logs shows browser-like strings — indicative of token reuse outside the agent runtime environment.

Evidence: Capture and preserve: (1) IdP audit logs for the past 90 days scoped to OAuth token issuance events, specifically `token.grant` and `app.oauth2.token.grant.implicit` event types (Okta) or `Sign-in activity` and `Audit logs > Application` categories (Entra ID). (2) API gateway or reverse-proxy access logs for HR data endpoints, code repository APIs (GitHub/GitLab `/api/v4/` paths), and CRM APIs — filter for requests where `Authorization: Bearer` header is present and `User-Agent` does not match your agent framework's declared string. (3) Raw JWT payloads from captured requests — the absence of agent-context claims (`agent_id`, `originating_user`, `delegation_chain`) in the JWT body is itself a forensic finding confirming the identity gap described in this advisory.

Step 3: Eradication — Implement compensating controls at the application layer. Add custom JWT claims or metadata headers to agent requests to carry agent instance ID, originating user context, and agent-user relationship. Enforce these claims in authorization middleware so any request lacking agent context is rejected. Where possible, issue agent-specific service accounts with scopes limited to named resources, not broad API permissions. Reference: NIST AC-3 (Access Enforcement), NIST AC-6 (Least Privilege), D3-UAP (User Account Permissions), D3-CH (Credential Hardening).

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST AC-3 (Access Enforcement), NIST AC-6 (Least Privilege), NIST AC-2 (Account Management)

Compensating: Without an enterprise identity platform, implement claim enforcement in application middleware using a lightweight JWT validation library. In Python (FastAPI/Flask), add a dependency that inspects decoded JWT payloads for a required `agent_id` claim: `if 'agent_id' not in token_claims: raise HTTPException(403, 'Agent context claim missing')`. For MCP-based agents, configure the MCP server's tool invocation handler to require an `x-agent-instance-id` HTTP header and reject requests missing it with HTTP 403 before any tool execution. Document

the claim schema (`agent_id`, `on_behalf_of_user`, `agent_framework`, `scope_justification`) as an internal standard and apply it to every new agent service account registration.

Evidence: BEFORE modifying authorization middleware or rotating service accounts, capture: (1) Current middleware configuration files and any API gateway policy definitions (e.g., AWS API Gateway resource policies, Kong plugin configs, NGINX auth_request blocks) — these establish pre-change baseline. (2) A snapshot of all existing agent service account definitions from your IdP, including current OAuth scopes granted, to document what was remediated and what was removed. (3) For Claude Code and MCP deployments specifically, capture the MCP server's active tool registry and any persisted session state files (typically in `~/.mcp/` or the agent's working directory) before reconfiguring — these may contain cached credentials or session tokens that survive a service restart.

Step 4: Recovery — Validate that revised token issuance policies are producing auditable records distinguishing agent actions from human actions. Test that short-lived agent tokens expire and rotate as configured. Confirm that revoked legacy tokens are no longer accepted by downstream services. Monitor identity provider logs for anomalous token usage patterns for 30 days post-remediation. Reference: NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST AU-9 (Protection of Audit Information), D3-CRO (Credential Rotation).

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST AU-6 (Audit Record Review, Analysis, And Reporting), NIST AU-9 (Protection Of Audit Information), NIST AU-3 (Content Of Audit Records)

Compensating: Automate token expiry validation with a scheduled script: use `curl -s -X POST -d 'token=&token_type_hint=access_token'` and parse the `active` field — any agent token returning `true` beyond its configured `exp` window indicates a rotation failure. Schedule this check as a daily cron job and log results to a tamper-evident append-only file (`>> /var/log/agent_token_audit.log`). For downstream service validation, replay a revoked token against each API endpoint that agent accounts previously accessed and confirm HTTP 401 responses — document each test with timestamp, endpoint, token hash (not full token), and response code.

Evidence: During recovery validation, preserve: (1) IdP token introspection responses for each newly issued agent token, confirming presence of `agent_id` claim, correct `exp` window, and restricted scope list — these serve as the post-remediation baseline. (2) API gateway access logs from the first 72 hours post-rotation, specifically filtering for HTTP 401 responses on endpoints previously accessed by agent service accounts — successful 401s confirm revocation propagated; any 200 responses on revoked token hashes require immediate escalation. (3) A before/after comparison of OAuth scope assignments for each agent service account, retained as a permanent remediation record.

Step 5: Post-Incident — Document the agent identity gap as a standing control deficiency in your risk register. Assign ownership to track the MCP authentication specification and any emerging IETF or NIST guidance on agentic identity. Establish a policy requiring all new AI agent deployments to include a documented identity model, minimum-scope token definition, and audit logging plan before production authorization. Reference: NIST AC-1 (Policy and Procedures), NIST AC-5 (Separation of Duties), CIS 7.1 (Establish and Maintain a Vulnerability Management Process).

NIST Phase: Post Incident

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

Controls: NIST AC-1 (Policy And Procedures), NIST AC-5 (Separation Of Duties), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: For teams without a formal GRC platform, maintain the risk register entry in a version-controlled markdown file in a private repository — include fields for: deficiency description (agent identity gap per RFC 9068 / OAuth 2.1), affected systems (all AI agent deployments), current compensating controls, residual risk rating, owner, and next review date tied to MCP spec or IETF draft updates. Set a GitHub/GitLab repository watch on the MCP specification repository and subscribe to the IETF OAuth working group mailing list (oauth@ietf.org) to receive notification of agentic identity drafts. Create a checklist-gated merge/deploy requirement: no AI agent reaches production without a completed `agent-identity-review.md` covering identity model, minimum-scope definition, and

logging plan.

Evidence: Retain as permanent post-incident documentation: (1) The full pre-remediation OAuth token audit export (captured in Step 1) as the evidentiary baseline of the deficiency's scope. (2) The middleware enforcement change records and service account scope reduction logs from Step 3, timestamped and signed by the implementing engineer, as proof of compensating control implementation. (3) The 30-day post-remediation IdP monitoring log summaries from Step 4, demonstrating no anomalous agent token reuse was observed — or, if anomalies were found, the escalation records generated. These three artifacts together constitute the evidence package for any audit or regulatory inquiry into AI agent identity governance.

Detection Guidance

Query your identity provider's token issuance logs for service principals or non-human accounts that have been granted OAuth scopes touching sensitive resource APIs (HR systems, code repositories, customer data stores). Flag any token with a lifetime exceeding 1 hour issued to a non-human principal. In SIEM, correlate OAuth access events against expected agent behavior baselines: a single agent instance accessing resources from multiple source IPs or presenting inconsistent user-agent strings is a strong behavioral indicator of token theft or misuse (T1550.001). Review audit logs for sequences matching T1134 (Access Token Manipulation): look for token refresh requests that alter scope or audience claims mid-session. Because current JWT structures carry no standardized agent-identity claims (CWE-1270), legitimate agent activity and adversarial token abuse will appear identical in standard logs; detection depends on behavioral anomaly detection, not claim inspection. Enable AU-12-compliant audit record generation across all systems that accept OAuth tokens from agent principals, and ensure records capture source identity, timestamp, resource accessed, and action performed per NIST AU-3. Where CrowdStrike Falcon Identity Security is deployed, apply vendor guidance for continuous identity monitoring of AI agent service accounts. Note: References to internal control frameworks (e.g., D3-LAM, D3-SFA) are mapped to NIST equivalents in parentheses for external reference.

Framework Mappings

MITRE-ATTACK

- **T1098** — Account Manipulation
- **T1550.001** — Application Access Token
- **T1134** — Access Token Manipulation
- **T1548** — Abuse Elevation Control Mechanism
- **T1078** — Valid Accounts
- **T1078.004** — Cloud Accounts

NIST-800-53R5

- **AC-6** — Least Privilege
- **CM-6** — Configuration Settings
- **AC-2** — Account Management
- **IA-2** — Identification and Authentication (Organizational Users)
- **IA-5** — Authenticator Management
- **IA-8** — Identification and Authentication (Non-Organizational Users)

- **AC-3** — Access Enforcement
- **AT-2** — Literacy Training and Awareness

OWASP-TOP10-2021

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

CIS-V8

- **6.3** — Require MFA for Externally-Exposed Applications
- **6.4** — Require MFA for Remote Network Access
- **6.5** — Require MFA for Administrative Access
- **6.1** — Establish an Access Granting Process
- **6.2** — Establish an Access Revoking Process
- **5.4** — Restrict Administrator Privileges to Dedicated Administrator Accounts
- **6.8** — Define and Maintain Role-Based Access Control
- **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
- **CC9.2** — Manages risks associated with vendors and business partners

HIPAA-SECURITY

- **164.312(d)** — Person or Entity Authentication
- **164.312(a)(1)** — Access Control

ISO-27001-2022

- **A.8.8** — Management of technical vulnerabilities
- **A.5.21** — Managing information security in the ICT supply chain

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1098	Account Manipulation	Persistence
T1550.001	Application Access Token	Defense-Evasion
T1134	Access Token Manipulation	Defense-Evasion
T1548	Abuse Elevation Control Mechanism	Privilege-Escalation
T1078	Valid Accounts	Defense-Evasion
T1078.004	Cloud Accounts	Defense-Evasion

Sources

Source	URL	Tier
Blog	https://www.crowdstrike.com/en-us/blog/the-identity-problem-hiding-...	T3
	https://www.crowdstrike.com/en-us/blog/crowdstrike-announces-contin...	T3
	https://www.crowdstrike.com/en-us/blog/crowdstrike-zscaler-bring-co...	T3
	https://hackernoon.com/7-security-problems-nobody-is-solving-in-the...	T3
Securing AI Agents: The Future of MCP Authentication & Authorization	https://mlops.community/blog/securing-ai-agents-the-future-of-mcp-a...	T3

Generated 2026-06-28 15:15 UTC by TJS Security Command Center