INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-06-26 06:30 UTC

OAuth Was Never Built for AI Agents: The Identity Gap Threatening Enterprise Agentic Deployments

SECURITY ANALYSIS | HIGH | CVSS 7.5

SCC Item ID	SCC-STY-2026-0276
Type	Security Analysis
Severity	HIGH
CVSS Base Score	7.5
Affected Products	OAuth 2.1 / JWT (RFC 9068) implementations broadly; enterprise IAM stacks issuing tokens to AI agents; MCP-compatible AI agents (e.g., Claude Code, autonomous HR/workflow agents); CrowdStrike Falcon Identity Protection; any agentic deployment lacking agent-specific identity fields
Discovery Source	Rss:T1 Threatintel

Executive Summary

Enterprise AI agent deployments are scaling into an identity framework, OAuth 2.1 and JWT, that was never designed to represent them. Because current token standards carry no fields for agent instance identity, the delegating user, or the relationship between them, downstream systems cannot distinguish an agent's actions from a human's, cannot enforce least-privilege controls scoped to the agent, and cannot produce audit trails attributable to a specific agent. With Gartner projecting 40% of enterprise applications embedding AI agents by end of 2026, this structural gap is expanding faster than any available standard can close it, and exploitation requires no novel technique: existing credential abuse and lateral movement playbooks apply directly to over-permissioned agent tokens.

Technical Analysis

The problem is architectural, not incidental. OAuth 2.1 and JWT-formatted access tokens defined under RFC 9068 were designed for delegated human authorization. They carry claims for the subject (sub), issuer (iss), audience (aud), and scope, but no standardized fields for agent instance identity, the identity of the human principal who delegated authority, or the bounded scope of what the agent is permitted to do on that principal's behalf. When an enterprise deploys an AI agent, such as an autonomous HR workflow bot or a coding assistant like Claude Code operating via the Model Context Protocol (MCP), it typically inherits a token scoped to the user or service account that instantiated it. That token, once issued, looks identical to any other OAuth bearer token

to every downstream system the agent touches.

This creates four compounding weaknesses. First, excessive privilege through ambient token inheritance (CWE-269, CWE-732): the agent receives whatever permissions the issuing identity held, not the narrower set required for its specific task. Second, audit trail failure (CWE-1318): because no agent-specific claim exists in the token, API logs, data store access records, and SIEM events record actions as originating from the human user or service account, not the agent instance. Incident responders and compliance auditors cannot reconstruct what the agent did versus what the human did. Third, missing authorization checks on agent-initiated requests (CWE-862): downstream APIs have no signal to apply agent-specific policy, even if an organization wanted to implement one. Fourth, the authentication gap itself (CWE-287): in many deployments, agent identity is never formally established at all.

The MITRE ATT&CK techniques this gap enables are not theoretical. T1078 (Valid Accounts) and T1550.001 (Application Access Token abuse) map directly to a scenario where a malicious agent, or an attacker who has compromised an agent runtime, uses inherited tokens to move laterally or exfiltrate data with no behavioral anomaly visible to the IAM layer. T1606 and T1606.001 (Forge Web Credentials) describe the upgrade path: an attacker who understands the token structure can craft or modify tokens that downstream systems will accept, because no agent-specific validation exists to reject them. T1134 and T1134.001 (Token Impersonation and Token Theft) describe the persistence play: once an agent token is obtained, it can be reused at scale. T1548 (Abuse Elevation Control Mechanism) and T1087 (Account Discovery) round out the lateral movement picture.

CrowdStrike has publicly characterized this problem and announced continuous identity monitoring for AI agents within Falcon Identity Protection as a partial mitigation, the first major commercial detection capability specifically targeting agent identity behavior. The IETF has no finalized RFC addressing agent-specific token claims as of this writing. Work is underway in identity and standards communities, as discussed at Identiverse 2026, but no consensus standard is close to ratification. Recorded Future's enterprise AI risk research corroborates the threat surface characterization. The gap between deployment velocity and standards maturity is not shrinking.

Action Checklist

1. Step 1: Assess exposure, audit all AI agent deployments to determine which OAuth tokens they hold, what scopes those tokens carry, and whether any agent is operating with a token scoped to a human user or a broadly-permissioned service account
2. Step 2: Review controls, apply NIST AC-6 (Least Privilege) to every agent identity: reduce token scopes to the minimum required for the specific agent task; apply NIST AC-3 (Access Enforcement) to verify downstream APIs enforce authorization checks on agent-sourced requests; verify CIS 5.1 (Establish and Maintain an Inventory of Accounts) covers agent identities, not only human accounts
3. Step 3: Enforce MFA and dedicated accounts, align with CIS 6.5 (Require MFA for Administrative Access) and CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts); agent service accounts should be isolated from human user accounts and carry no administrative privilege by default
4. Step 4: Instrument logging for agent attribution, implement NIST AU-3 (Content of Audit Records) and AU-12 (Audit Record Generation) with custom fields or proxy-layer annotations that tag agent-initiated requests at the API gateway or identity proxy; without this, audit logs will not support incident reconstruction or compliance review

5. Step 5: Deploy or evaluate agent-aware identity monitoring, review CrowdStrike Falcon Identity Protection's continuous agent identity monitoring capability or equivalent; map coverage to MITRE D3-LAM (Local Account Monitoring) and D3-UAP (User Account Permissions) to detect token abuse and out-of-scope behavior by agent instances

6. Step 6: Update threat model, add T1078, T1550.001, and T1606 to your threat register under an 'AI Agent Token Abuse' scenario; document the absence of agent-specific token standards as a sustained architectural risk until an IETF standard is ratified

7. Step 7: Monitor standards developments, track IETF working group progress on agent-specific JWT claims and Identiverse/identity community outputs; assign an IAM or GRC owner to update token issuance policies when a standard reaches ratification

IR / Forensic Enrichment

Triage Priority	URGENT
Escalation Criteria	Escalate immediately to CISO and legal counsel if audit output from Step 1 reveals any AI agent currently holding a token scoped to a human user identity with access to PII, PHI, or financial data, as this condition may constitute an unauthorized data access event triggering breach notification obligations under GDPR, HIPAA, or applicable state law.
Recovery Notes	After scope reduction and account isolation are complete, re-run the token introspection audit from Step 1 to confirm no agent client_id retains scopes beyond its documented minimum; treat any scope regression as an active incident indicator. Monitor IdP audit logs and the proxy-layer agent attribution logs (Step 4) continuously for 30 days post-remediation, specifically watching for agent service accounts attempting to re-acquire revoked scopes via dynamic client registration or token exchange flows. Recovery is not complete until agent-attributable logging is operational and at least one detection rule (Step 5) has produced a verified true-positive alert in a test scenario.
Forensic Artifacts	IdP OAuth token grant logs: export all token issuance events filtered by agent client_id values, preserving the full JWT payload (sub, aud, scope, azp, iat, exp claims) — these establish which agent held which scopes and for how long prior to remediation | API gateway access logs annotated with JWT azp and client_id fields: the pre-instrumentation logs (before Step 4) will show agent-sourced requests recorded under human sub claims, which is the primary forensic indicator of the identity attribution gap in this threat | MCP server or agent orchestration platform session logs: for Claude Code or equivalent MCP-compatible agents, the orchestration layer logs will show tool invocations, resource targets, and delegation chain — these are specific to the agentic deployment model and not present in standard OAuth audit trails | CrowdStrike Falcon Identity Protection identity risk event timeline (if deployed): filter on service principal anomaly detections and token-based lateral movement indicators tied to agent account identities to establish a behavioral timeline of any token abuse preceding discovery | In-memory JWT token cache artifacts from agent process memory: on Linux agent hosts, `/proc//maps` and `/proc//mem` may contain cached bearer tokens in Base64 form; on Windows, a user-mode memory dump of the agent orchestration process (e.g., obtained via `procdump.exe -ma `) may contain JWT strings recoverable with a YARA rule targeting the Base64 JWT header pattern — these are volatile and must be captured before any process termination or host isolation

Per-Action IR Details

Step 1: Assess exposure — audit all AI agent deployments to determine which OAuth tokens they hold, what scopes those tokens carry, and whether any agent is operating with a token scoped to a human user or a broadly-permissioned service account

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: establishing asset visibility and understanding the attack surface before an incident occurs

Controls: NIST AC-2 (Account Management), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), CIS 5.1 (Establish and Maintain an Inventory of Accounts)

Compensating: Use `az ad sp list --all` (Azure) or `gcloud iam service-accounts list` (GCP) to enumerate service principals and service accounts, then cross-reference with OAuth token introspection endpoints (`POST /oauth/introspect`) to dump active scopes per agent identity. For on-prem or generic OAuth2 stacks, run `curl -X POST -d 'token='` for each known agent credential. Build a CSV mapping agent name → client_id → granted scopes → associated human user (if any). Flag any entry where `sub` claim resolves to a human UPN or where scopes include write/admin-level permissions.

Evidence: Before any scope changes or account modifications, snapshot current token state: capture active token grants via the IdP's admin API (e.g., Azure AD `GET /v1.0/oauth2PermissionGrants`, Okta `GET /api/v1/authorizationServers/{id}/policies`), export JWT payloads by decoding active bearer tokens with `jwt decode ` (jwt-cli) to preserve the `sub`, `aud`, `scope`, `azp`, and `client_id` claims at the time of audit. This baseline is your pre-remediation record of over-permissioned agent identities.

Step 2: Review controls — apply NIST AC-6 (Least Privilege) to every agent identity: reduce token scopes to the minimum required for the specific agent task; apply NIST AC-3 (Access Enforcement) to verify downstream APIs enforce authorization checks on agent-sourced requests; verify CIS 5.1 (Establish and Maintain an Inventory of Accounts) covers agent identities, not only human accounts

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment Strategy: limiting the blast radius of over-permissioned agent tokens by reducing granted scopes and enforcing authorization boundaries at downstream APIs

Controls: NIST AC-3 (Access Enforcement), NIST AC-6 (Least Privilege), CIS 5.1 (Establish and Maintain an Inventory of Accounts)

Compensating: For API gateways without native agent-scope enforcement, deploy an NGINX or Envoy reverse proxy with Lua/WASM policy that inspects the JWT `scope` claim on inbound requests and rejects any agent token presenting scopes beyond a defined allowlist. Use `opa eval` (Open Policy Agent, free) with a Rego policy that checks `input.token.scope` against a per-agent scope map stored in a flat JSON file. For scope reduction on Azure AD, use `Remove-MgOauth2PermissionGrant` (Microsoft Graph PowerShell) to revoke excess delegated permissions without requiring enterprise tooling.

Evidence: Before revoking or reducing any token scope — which terminates live agent sessions and destroys in-flight token state — capture: (1) full JWT decode of all active agent tokens (`scope`, `exp`, `iat`, `azp`, `sub` claims); (2) IdP audit log export covering agent token issuance events for the prior 90 days (Azure AD Sign-in logs filtered on `appDisplayName` matching known agent client IDs, or equivalent IdP audit trail); (3) API gateway access logs filtered on the agent's `client_id` to establish the scope of resource access prior to restriction. These records are necessary to determine whether any agent already acted outside its intended permission boundary.

Step 3: Enforce MFA and dedicated accounts — align with CIS 6.5 (Require MFA for Administrative Access) and CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts); agent service accounts should be isolated from human user accounts and carry no administrative privilege by default

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment Strategy: isolating agent identities from human identity plane to prevent lateral privilege escalation via token reuse or delegation chain abuse

Controls: NIST AC-5 (Separation of Duties), NIST AC-6 (Least Privilege), CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts), CIS 6.5 (Require MFA for Administrative Access)

Compensating: Use your IdP's group policy to place all agent service accounts into a dedicated non-interactive OU or group with a Conditional Access policy that blocks interactive login entirely (Azure AD: set `signInAudience` to service-principal only; block user-assignable roles). For MFA on administrative agent orchestration consoles, enforce TOTP via a free RADIUS server (FreeRADIUS) fronting the management plane. Audit group membership weekly with `Get-MgGroupMember` (Graph PowerShell) or `aws iam list-groups-for-user` to detect agent service accounts added to privileged groups.

Evidence: Before isolating agent service accounts from the human identity plane (which will terminate any active sessions where an agent is co-mingled with a human identity), capture: (1) IdP directory export of all accounts with `servicePrincipalType` or equivalent flag, cross-referenced against role assignments to identify human-privileged roles held by agent identities; (2) active session list for each agent account (`Get-MgUser -UserId | Get-MgUserAuthenticationSession` or equivalent) to document sessions that will be terminated; (3) MCP server or orchestration platform session logs showing which agent instances were authenticated at the time of isolation.

Step 4: Instrument logging for agent attribution — implement NIST AU-3 (Content of Audit Records) and AU-12 (Audit Record Generation) with custom fields or proxy-layer annotations that tag agent-initiated requests at the API gateway or identity proxy; without this, audit logs will not support incident reconstruction or compliance review

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: establishing attributable audit trails for agent-initiated API actions, which are currently indistinguishable from human actions in standard OAuth/JWT audit logs

Controls: NIST AU-3 (Content Of Audit Records), NIST AU-12 (Audit Record Generation), CIS 8.2 (Collect Audit Logs)

Compensating: Deploy a lightweight API gateway proxy (Traefik or Caddy, both free) that extracts the JWT `azp` (authorized party) and `client_id` claims from every inbound bearer token and appends them as structured log fields (`agent_id`, `delegating_user`, `token_scope`) to the access log. Ship these logs to a free ELK stack or Loki instance. Write a Sigma rule detecting any API request where `agent_id` is non-null but `delegating_user` is null or where `token_scope` contains write/delete verbs against sensitive resource paths. This gives post-hoc attribution without a commercial SIEM.

Evidence: This step does not alter live system state but establishes the logging baseline. Before deploying proxy-layer annotation changes (which may briefly interrupt agent traffic), capture a pre-change sample of raw API gateway logs to document the current attribution gap — specifically, entries where the JWT `sub` claim resolves to a human UPN even though the request originated from an automated agent process. This sample serves as evidence of the pre-remediation blind spot in any future compliance or incident review.

Step 5: Deploy or evaluate agent-aware identity monitoring — review CrowdStrike Falcon Identity Protection's continuous agent identity monitoring capability or equivalent; map coverage to MITRE D3-LAM (Local Account Monitoring) and D3-UAP (User Account Permissions) to detect token abuse and out-of-scope behavior by agent instances

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: implementing continuous monitoring of agent identity behavior to surface token abuse, scope anomalies, and out-of-baseline resource access before they escalate

Controls: NIST AU-6 (Audit Record Review, Analysis, And Reporting), MITRE D3FEND D3-LAM (Local Account Monitoring), MITRE D3FEND D3-UAP (User Account Permissions)

Compensating: Without CrowdStrike Falcon Identity Protection, deploy osquery with the `user_groups`, `logged_in_users`, and `processes` tables on agent host nodes, scheduled at 5-minute intervals, to detect unexpected interactive sessions or privilege changes on agent service accounts. Write a YARA rule targeting in-memory JWT structures (look for Base64-encoded header `eyJhbGciOiJSUzI1NiJ9` patterns in process memory) and run it against agent process memory snapshots using `yara -p 4 rule.yar /proc//mem` on Linux hosts. For Windows-based MCP agent deployments, use Sysmon Event ID 10 (ProcessAccess) to flag any process attempting to read the memory of the agent orchestration process where tokens may be cached.

Evidence: Before deploying any new monitoring agent or reconfiguring existing identity monitoring (which may clear event queues or reset behavioral baselines), capture: (1) current osquery or EDR behavioral baseline for each agent service account — specifically the set of resource endpoints accessed, request volumes, and time-of-day patterns over the prior 30 days; (2) IdP risk event logs (Azure AD Identity Protection risk detections, or equivalent) filtered on agent client IDs to identify any already-flagged anomalous token usage; (3) network flow records (NetFlow/IPFIX or `ss -tnp` output from agent hosts) showing established connections from agent processes at the time of baseline capture.

Step 6: Update threat model — add T1078, T1550.001, and T1606 to your threat register under an 'AI Agent Token Abuse' scenario; document the absence of agent-specific token standards as a sustained architectural risk until an IETF standard is ratified

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: updating organizational threat models and risk registers based on identified architectural gaps, consistent with lessons-learned and continuous improvement objectives

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

Compensating: Maintain the threat register as a version-controlled Markdown or JSON file in a Git repository. For each ATT&CK technique referenced (T1078 Valid Accounts, T1550.001 Application Access Token, T1606 Forge Web Credentials), create a corresponding Sigma rule stub targeting the agent-specific log fields instrumented in Step 4, even if detection is initially incomplete. Use MITRE ATT&CK Navigator (free, browser-based) to annotate your coverage layer, marking these techniques as 'partial' until agent-attributable logging from Step 4 is fully operational. Document the IETF draft status (e.g., draft-ietf-oauth-identity-chaining) as a dated risk acceptance entry.

Evidence: No live system state is altered by this step; no volatile evidence capture is required. However, attach to the threat register entry the token audit output from Step 1, the scope reduction records from Step 2, and the pre-instrumentation log gap sample from Step 4 as supporting evidence that the architectural risk is real and currently observable in your environment — not merely theoretical.

Step 7: Monitor standards developments — track IETF working group progress on agent-specific JWT claims and Identiverse/identity community outputs; assign an IAM or GRC owner to update token issuance policies when a standard reaches ratification

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: assigning organizational ownership for sustained architectural risk tracking and policy update readiness as the IETF agent identity standard matures

Controls: NIST AC-1 (Policy And Procedures), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Subscribe the assigned IAM/GRC owner to the IETF OAuth working group mailing list (free, ietf.org/mailman/listinfo/oauth) and configure an RSS feed alert on the IETF Datatracker for drafts matching 'agent' or 'identity-chaining' keywords. Set a calendar-triggered quarterly review task to assess whether any new IETF draft or RFC has been published that defines agent-specific JWT claims (e.g., `act`, `may_act`, or a new agent instance identifier claim), and tie that review to a policy update gate in the token issuance configuration of your IdP.

Evidence: No live system state is altered by this step; no volatile evidence capture is required. Document the current IETF draft version number and publication date at the time this step is completed, so the IAM/GRC owner has a versioned baseline from which to measure standards progression. Attach this record to the risk register entry created in Step 6.

Detection Guidance

Detection is constrained by the core problem: current logs do not natively distinguish agent actions from human actions. Compensating controls focus on behavioral anomaly and proxy-layer attribution.

Audit log review: Query identity and API access logs for service accounts or user tokens that are active outside normal business hours, initiating unusually high request volumes, or accessing resource combinations

inconsistent with the account's defined role. These patterns may indicate an agent operating with inherited credentials. NIST AU-6 (Audit Record Review, Analysis, and Reporting) applies directly. Flag accounts where the same token is used across multiple sessions or source IPs simultaneously, consistent with T1550.001 token reuse.

Agent inventory gaps: Run CIS 1.1 (Enterprise Asset Inventory) and CIS 5.1 (Account Inventory) checks specifically against AI agent service accounts. Any agent identity not formally registered, scoped, and documented in your IAM system is a blind spot. CrowdStrike's continuous identity monitoring for AI agents is designed to surface unregistered or anomalously behaving agent identities in Falcon Identity Protection.

Token scope audits: Pull all active OAuth tokens from your authorization server and flag any token held by a non-human identity with scopes exceeding read access to a single data class. Tokens carrying write, delete, or admin scopes assigned to agent accounts should trigger immediate review.

MCP-specific monitoring: If your environment includes MCP-compatible agents (Claude Code or similar), instrument the MCP server layer to log tool invocations with the agent instance identifier appended. Absent native token fields, this is currently the most reliable attribution method available.

Behavioral hunting hypothesis: Hunt for T1087 (Account Discovery) activity originating from service accounts, specifically directory enumeration or permission probing that no scheduled job or known automation task should be performing. Agent runtimes compromised by a malicious prompt or supply chain modification may exhibit this pattern as a precursor to lateral movement.

D3FEND countermeasures: Apply D3-UAP (User Account Permissions) to enforce strict permission boundaries on agent accounts; D3-LAM (Local Account Monitoring) to surface anomalous agent account behavior; D3-CRO (Credential Rotation) to limit the window of exposure for any compromised agent token.

Framework Mappings

MITRE-ATTACK

- **T1606** — Forge Web Credentials
- **T1078** — Valid Accounts
- **T1550.001** — Application Access Token
- **T1548** — Abuse Elevation Control Mechanism
- **T1087** — Account Discovery
- **T1134.001** — Token Impersonation/Theft
- **T1606.001** — Web Cookies
- **T1134** — Access Token Manipulation

NIST-800-53R5

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

- **AC-3** — Access Enforcement
- **SI-4** — System Monitoring

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
- **5.4** — Restrict Administrator Privileges to Dedicated Administrator Accounts
- **6.8** — Define and Maintain Role-Based Access Control
- **3.3** — Configure Data Access Control Lists
- **6.1** — Establish an Access Granting 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

NIST-CSF-2

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

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1606	Forge Web Credentials	Credential-Access
T1078	Valid Accounts	Defense-Evasion
T1550.001	Application Access Token	Defense-Evasion
T1548	Abuse Elevation Control Mechanism	Privilege-Escalation
T1087	Account Discovery	Discovery
T1134.001	Token Impersonation/Theft	Defense-Evasion
T1606.001	Web Cookies	Credential-Access
T1134	Access Token Manipulation	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://securityboulevard.com/2026/06/identiverse-2026-the-challeng...	T3
	https://www.recordedfuture.com/research/emerging-enterprise-securit...	T3
The Identity Problem Hiding in AI Agent Deployments | CrowdStrike	https://www.crowdstrike.com/content/crowdstrike-www/locale-sites/us...	T3

Generated 2026-06-26 06:30 UTC by TJS Security Command Center