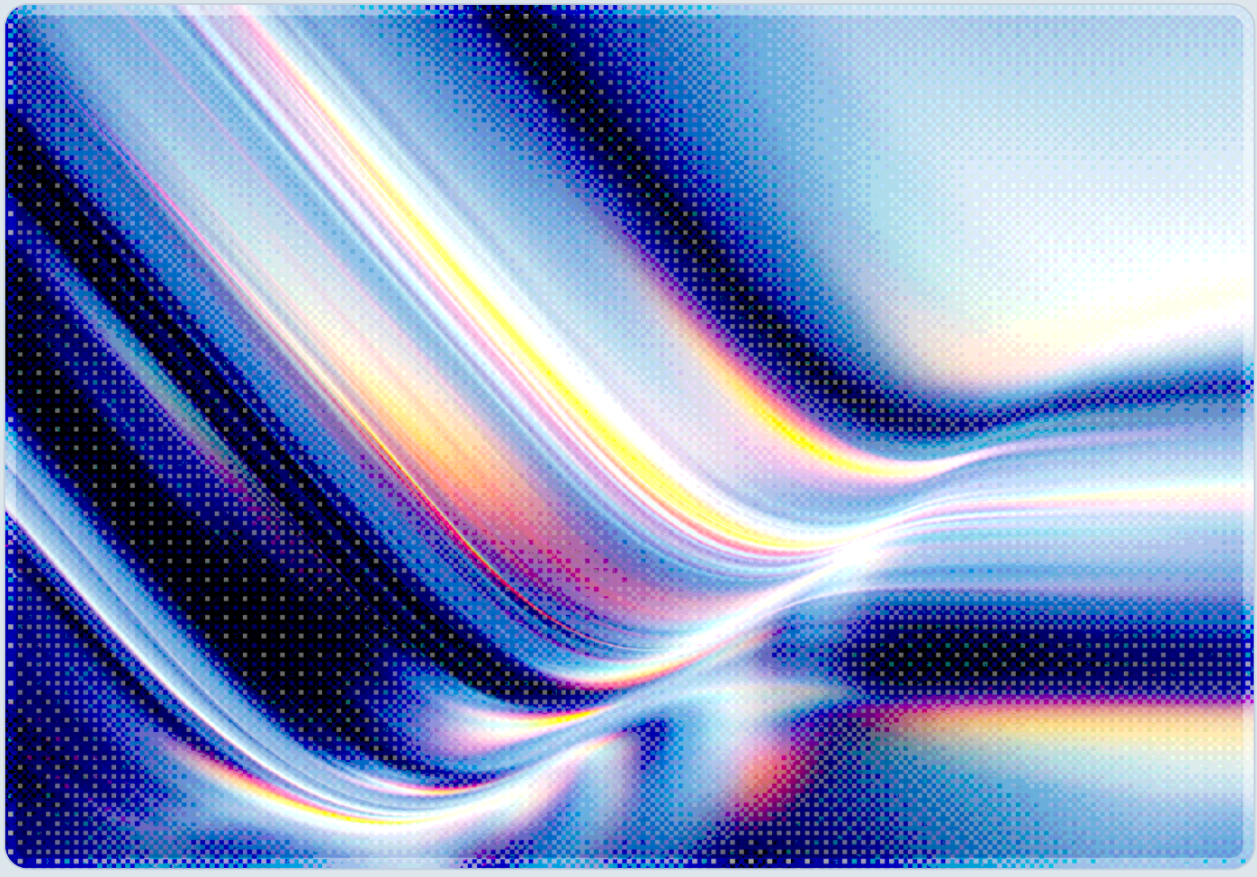
CSAI Foundation | Cloud Security Alliance

Persistent OAuth Tokens in AI-Integrated Environments

The Access Path IAM Doesn't See

2026-05-06

AI-assisted Rapid Research



CSAI

CSA cloud security alliance®

Key Takeaways

The rapid integration of AI coding assistants, agentic platforms, and productivity tools into enterprise environments has introduced a category of persistent access that conventional IAM deployments are frequently ill-positioned to detect. The following findings summarize the core risk profile addressed in this note:

- AI tools commonly acquire OAuth tokens—including long-lived refresh tokens—that remain valid independent of the user's current IAM posture, surviving password resets, role changes, and account suspensions in environments where identity providers do not include refresh token revocation as part of credential-change flows [6].
- A critical vulnerability in OpenAI Codex (disclosed December 16, 2025) demonstrated that command injection via unsanitized repository branch names could exfiltrate a developer's GitHub OAuth token in cleartext to an attacker-controlled server [1][2].
- AI-enabled device code phishing campaigns, including a 2026 campaign using the "EvilTokens" platform-as-a-service, have compromised more than 340 Microsoft 365 organizations across five countries by exploiting the OAuth 2.0 device authorization flow [3] [4].
- Non-human identities—including AI agents—now outnumber human identities in many enterprises by ratios that make manual oversight untenable, with CyberArk's 2025 Identity Security Landscape survey documenting machine-to-human identity ratios as high as 82:1 in organizations with significant SaaS and automation footprints [21].
- When a user's access is revoked in an identity provider, AI agent connectors operating on previously issued tokens frequently continue operating without re-evaluation, creating persistent access paths that are invisible to standard IAM tooling [6].
- Addressing this risk requires both technical controls—short-lived tokens, scope minimization, behavioral monitoring—and governance changes to bring AI tool integrations under the same lifecycle management as human accounts.

Background

OAuth 2.0 has become the de facto authorization framework underpinning enterprise SaaS integrations, and AI tools have adopted it enthusiastically. When a developer connects GitHub Copilot to their code repository, or an analyst authorizes a productivity AI to read their calendar and email, the result is not merely a one-time connection: the tool typically receives both a short-lived access token and a long-lived refresh token. The refresh token—often valid for weeks, months, or indefinitely depending on vendor configuration—allows the AI tool to automatically re-acquire access tokens without further user interaction [6].

This persistence is by design. OAuth 2.0's refresh token mechanism exists precisely to enable applications to maintain access over time without requiring repeated authentication from the user. For consumer applications, this is an intended usability feature. In enterprise environments with AI tools that interact with sensitive systems—source code repositories, cloud management APIs, communication platforms, and document stores—it represents an access path that operates outside the normal rhythm of IAM governance.

Traditional IAM systems are designed around human principals: provisioning is tied to onboarding, de-provisioning to offboarding, and access reviews happen at regular intervals keyed to employee records. When an employee's permissions are reduced or their account suspended, the identity provider enforces those changes at the authentication layer. OAuth tokens, however, encode authorization state at the time of issuance and carry it forward. An access token does not consult the identity provider each time it is used. A refresh token generates new access tokens until it is explicitly revoked or expires—and explicit revocation requires the security team to know the token exists in the first place [6].

The risk surface expands as AI tools become more capable and take on more autonomous roles across an organization's systems. Agentic AI systems—those capable of taking autonomous action across multiple services—may hold dozens of OAuth grants across an organization's SaaS portfolio. In many organizations, these grants are provisioned by individual employees during onboarding or experimentation, recorded in no centralized registry, and rarely revisited. The result is an unmanaged access layer that accumulates over time, scoped to permissions that may have been appropriate at the moment of consent but reflect none of the access changes that occur over the employee's tenure.

Security Analysis

The Token Acquisition Surface

AI tools acquire OAuth tokens through several mechanisms, each with distinct risk characteristics. The most common is the standard authorization code flow, in which the user explicitly consents in a browser session. This is the most visible path and the one most likely to be captured in audit logs if the identity provider supports OAuth consent event logging. A second mechanism—the device code flow—was designed for input-constrained devices like smart TVs and CLI tools, but has become a widely used acquisition method for AI development tools such as GitHub Copilot, the Claude CLI, and various agentic frameworks. In the device code flow, the user enters a short code in a browser while the application polls an authorization endpoint; once authentication completes, the application receives tokens without the user's browser ever touching the application's domain [23].

The device code flow is particularly susceptible to social engineering. Attackers can present legitimate-looking device code prompts to users, capturing tokens the moment the user authenticates. The EvilTokens phishing-as-a-service platform, which surfaced publicly in February 2026 and was used to compromise over 340 Microsoft 365 organizations, automated this process at scale: large language models generated contextually tailored lure content, automated polling infrastructure completed device authorization flows within the standard 15-minute expiration window before targets recognized and canceled the request, and within minutes of a successful authentication, attackers registered new devices to acquire Primary Refresh Tokens for long-term persistence [3][4][15]. The AI-enabled attack infrastructure specifically targeted the same OAuth mechanism that AI productivity tools use legitimately, exploiting the familiarity of those workflows to make malicious authentication prompts appear routine.

A third acquisition path—and the most concerning from a supply chain perspective—involves AI platforms that accept repository or data source credentials as part of their operating environment. The Codex vulnerability disclosed by BeyondTrust Phantom Labs on December 16, 2025 illustrates this plainly. Codex received a GitHub OAuth token scoped to all repositories the developer had authorized. When a malicious repository branch name containing injected shell commands was processed during environment setup, those commands executed in the Codex container context and exfiltrated the OAuth token in cleartext to a remote attacker server [1][2]. OpenAI classified the issue as Critical Priority 1, with full remediation completed approximately 47 days after disclosure in February 2026—a window during which the vulnerability was known but active [1][2]. The Codex incident illustrates a pattern documented across multiple AI coding tools, in which the credential—not the model itself—is the primary attack target [14].

The IAM Visibility Gap

The persistence problem is not exclusively about tokens being stolen. It also encompasses tokens that function precisely as intended but outlive the access policies that should govern them. When a developer is offboarded, the de-provisioning workflow terminates their user account and removes their group memberships. What it typically does not do is enumerate every third-party OAuth grant that developer's account authorized and revoke those grants. Unless the organization has deployed a dedicated SaaS security posture management (SSPM) solution or a non-human identity (NHI) governance platform, there is no system of record for these grants. The AI tool continues to operate with the former employee's delegated permissions until the refresh token expires or the target service's own token rotation forces a re-authentication that will fail [5][6].

This visibility gap extends beyond offboarding. An employee's role changes, their access to a sensitive project ends, or a security policy is updated to restrict access to production systems—none of these events propagate automatically to existing OAuth grants held by AI tools. The OAuth connector does not re-evaluate scope on each API call; it presents the token it holds and the receiving service honors it, provided the token has not expired and has not been explicitly revoked [6]. Research on SaaS supply chain token management has found that without dedicated OAuth grant lifecycle management processes, unauthorized access may persist for extended periods after policy changes take effect, because no mechanism exists to connect IAM policy updates to OAuth grant revocation workflows [5].

The Non-Human Identity Scale Problem

The challenge is compounded by the sheer volume of machine identities now operating in enterprise environments. The term "non-human identity" (NHI) encompasses service accounts, API keys, bot accounts, and the OAuth grants held by AI tools. CyberArk's 2025 Identity Security Landscape survey found machine-to-human identity ratios of 82:1 in organizations with significant SaaS and automation footprints [21]. As AI agents proliferate—Gartner has estimated that 40% of enterprise applications will integrate task-specific AI agents by the end of 2026, up from less than 5 percent in 2025 [22]—this ratio will continue to grow.

The scale creates a governance paradox: the volume of NHIs makes manual review impractical, but automated solutions for AI agent identity governance are immature. Many organizations have invested in human identity lifecycle tooling but have not extended those investments to machine identities, let alone the newer class of agentic AI identities that combine autonomous decision-making with broad OAuth grants [8]. When an AI agent holds refresh tokens across a dozen SaaS systems, operates across multiple tenants, and was provisioned by an employee acting unilaterally, no single team owns the governance responsibility [7][9].

The Authentication–Authorization Conflation

A final structural issue compounds all of the above. AI tool integrations—and the security reviews commonly applied to them—tend to focus on the authentication question: is this agent who it claims to be? OAuth provides a technically sound answer to that question. The authorization question—should this agent, at this moment, with this scope, perform this action?—receives far less attention. Once a token is issued, the scope encoded in it governs authorization for its full validity period. There is no mechanism in standard OAuth 2.0 for a service to query the issuing identity provider mid-session to confirm that the originally granted scope still reflects current policy [17]. While some practitioners argue that standard OAuth mechanisms are sufficient for AI agent deployments [16], the persistent, broad-scoped grants that agentic systems typically hold make the absence of continuous authorization verification a meaningful risk.

This creates what may be called an authorization gap: the period between when access was correct and when it should have been revoked is invisible to the receiving service. For human users, behavioral anomaly detection and session management tools provide partial mitigation. For AI agents, which may legitimately perform unusual data access patterns as part of their function, these controls are harder to calibrate and less likely to be deployed [18][19].

Recommendations

Immediate Actions

Organizations should treat OAuth grant discovery as an urgent gap to close. Security teams should audit all OAuth applications authorized against major identity providers—Microsoft Entra, Google Workspace, Okta, and others—with particular attention to grants carrying broad scopes such as `repo`, `mail.read`, `files.readwrite.all`, and similar permissions. Many identity providers expose this data through admin APIs or governance dashboards, but it is rarely reviewed with the same rigor applied to user access reviews. The audit should specifically tag grants associated with known AI tools and agentic platforms, and any grants associated with accounts that have been offboarded should be revoked immediately.

For AI coding tools that connect to source code repositories, organizations should enforce the principle of least privilege at the OAuth scope level. Where tools offer fine-grained scoping—limiting access to specific repositories rather than the full organization—those restrictions should be applied by default. The

Codex incident demonstrated that a single exfiltrated token scoped to all authorized repositories becomes an organization-wide credential exposure, not merely an individual developer incident [1][2].

Finally, where AI tools support it, organizations should disable the device code flow for enterprise tenants or require conditional access policies that block device code authentication from unmanaged devices. The device code flow's design—authenticating on one device while a separate application polls for the result—is inherently resistant to phishing-resistant authentication factors, making it a preferred target for the credential-harvesting campaigns observed in early 2026 [4].

Short-Term Mitigations

Over the next 30 to 90 days, organizations should establish a continuous OAuth grant monitoring practice. This involves periodic enumeration of all active grants, comparison against a known-good baseline, and alerting when new high-privilege AI tool grants appear outside of an approved provisioning process. Several SaaS security posture management platforms now include NHI visibility as a core capability; where these tools are available, their NHI modules should be activated and tuned to flag AI tool integrations.

Token rotation policies should be reviewed and, where possible, tightened. OAuth refresh tokens that do not expire present indefinite risk windows. Many major identity providers allow administrators to configure refresh token lifetimes; reducing the maximum token lifetime forces re-authentication at intervals and narrows the gap between policy changes and effective access revocation. Pairing shorter token lifetimes with centralized token storage—rather than allowing each AI tool to store credentials in its own environment—reduces the attack surface for credential exfiltration of the type demonstrated in the Codex vulnerability [2][20].

Organizations running AI coding agents or multi-step agentic workflows should conduct a scoped threat model for each integration, evaluating the prompt injection surface in particular. The Codex vulnerability originated from unsanitized input (a branch name) reaching an execution context that held elevated credentials. Similar injection surfaces may exist wherever AI agents accept external content—pull request titles, issue descriptions, email subject lines, document names—and route it into prompts that have access to OAuth-bearing environment variables. Code reviews for agentic integrations should explicitly include an assessment of how externally-controlled strings reach execution paths with credential access [1][2].

Strategic Considerations

In the longer term, organizations should develop an AI tool authorization policy that treats OAuth grant provisioning for AI systems as a governed process, parallel to service account provisioning. This means requiring an approval workflow before an AI tool is authorized against production systems, recording the

grant in an inventory, assigning an owner, and including AI tool grants in periodic access reviews. The same lifecycle discipline applied to traditional service accounts should extend to AI agent identities, with the added consideration that AI agents may hold grants across multiple systems simultaneously, creating a blast radius (scope of potential impact) that exceeds that of a single service account.

The industry is beginning to develop technical standards that address the agent identity problem more directly. OAuth 2.0 Token Exchange (RFC 8693) and emerging work on agent-specific identity protocols provide mechanisms for scoped delegation—allowing an AI agent to operate with a subset of the user's permissions in a verifiable, auditable way, rather than with a copy of the user's full grant [17]. Organizations selecting AI platforms should evaluate vendor roadmaps for adoption of these mechanisms, and security architects should include delegation chain auditability as a requirement in procurement decisions.

Zero trust principles apply directly to this problem domain. The architectural goal is to eliminate the implicit trust that currently flows from a valid OAuth token to an authorized session. Continuous verification—checking not just that an agent presents a valid token, but that the action it is attempting is consistent with current policy and expected behavior—is the architectural direction that addresses the authentication–authorization gap at its root [9]. CSA's Agentic Trust Framework, published in early 2026, provides a governance architecture for applying zero trust principles to agentic AI environments and should be consulted when designing access control architectures for AI agent deployments [11].

CSA Resource Alignment

This research note connects to several active CSA frameworks and publications that provide implementation guidance for the controls described above.

MAESTRO (Multi-Agent Environment, Security, Threat, Risk, and Outcome) is CSA's seven-layer threat modeling framework for agentic AI systems [10]. Layer 3 (Agent Frameworks) and Layer 6 (Security and Compliance) are directly relevant to token management and identity governance in multi-agent deployments. MAESTRO's threat taxonomy includes credential exfiltration, scope abuse, and cross-agent privilege escalation—the precise attack classes illustrated by the Codex vulnerability and the EvilTokens campaign. CSA published applied guidance for MAESTRO threat modeling in CI/CD pipeline contexts in February 2026, which addresses OAuth token handling in agent build and execution environments [12].

CSA Agentic AI Identity and Access Management is a purpose-built framework addressing the identity governance challenges unique to autonomous AI systems. It extends traditional IAM concepts to cover agent ownership chains, delegation patterns, ephemeral versus persistent agent identity, and multi-agent trust hierarchies. Organizations designing or procuring agentic AI platforms should use this framework to evaluate vendor identity models and identify governance gaps [9].

CSA Agentic Trust Framework provides zero trust governance architecture for AI agent deployments, directly addressing the continuous verification requirements identified in this note's strategic recommendations. The framework emphasizes that authentication alone—even when technically sound—is insufficient for governing autonomous agents that hold persistent, broad-scoped credentials [11].

CSA Cloud Controls Matrix (CCM) control domains IAM-01 through IAM-12 address identity lifecycle management and access control requirements that extend naturally to AI tool OAuth grants. The CCM's shared responsibility model is particularly relevant when AI platforms are SaaS-delivered: the organization retains responsibility for the grants it authorizes even when the platform itself is vendor-managed.

CSA AI Controls Matrix (AICM) provides a structured control framework covering AI-specific security domains. AICM controls in the data security and AI supply chain domains address the third-party OAuth integration risks documented in this note, including vendor token handling practices and supply chain compromise scenarios involving third-party SaaS OAuth integrations, such as the Salesloft-Drift breach in which a compromised integration token provided attackers sustained access to customer data [5][6].

CSA Labs published a complementary research note in March 2026 specifically addressing OAuth device code phishing against Microsoft 365 environments, which should be read alongside this note for organizations evaluating their exposure to the EvilTokens campaign and related threats [13].

References

[1] SecurityWeek. "Critical Vulnerability in OpenAI Codex Allowed GitHub Token Compromise." SecurityWeek, March 2026. (Retrospective analysis of the BeyondTrust disclosure.)

[2] BeyondTrust Phantom Labs. "OpenAI Codex Command Injection Vulnerability." BeyondTrust Blog, December 2025.

[3] The Hacker News. "Device Code Phishing Hits 340+ Microsoft 365 Orgs Across Five Countries via O Auth Abuse." The Hacker News, March 2026.

[4] Microsoft Security Blog. "Inside an AI-Enabled Device Code Phishing Campaign." Microsoft, April 2026.

[5] Palo Alto Networks Unit 42. "Trusted Connections, Hidden Risks: Token Management in the Third-Part y Supply Chain." Unit 42 Research, September 2025.

[6] Obsidian Security. "What Are OAuth Tokens? How It Works, and Its Vulnerabilities." Obsidian Security Blog, February 2026.

[7] Obsidian Security. "What Are Non-Human Identities? The Complete Guide to NHI Security." Obsidian Security Blog, February 2026.

[8] Oasis Security. "How 2025 Changed the Way We Think About Identity Security." Oasis Security Blog, December 2025.

[9] Cloud Security Alliance. "Agentic AI Identity and Access Management: A New Approach." CSA, August 2025.

[10] Cloud Security Alliance. "Agentic AI Threat Modeling Framework: MAESTRO." CSA Blog, February 2025.

[11] Cloud Security Alliance. "The Agentic Trust Framework: Zero Trust Governance for AI Agents." CSA Blog, February 2026.

[12] Cloud Security Alliance. "Applying MAESTRO to Real-World Agentic AI Threat Models: From Framew ork to CI/CD Pipeline." CSA Blog, February 2026.

[13] CSA Labs. "OAuth Device Code Phishing Hits 340+ Microsoft 365 Organizations." CSA Research Note, March 2026.

[14] VentureBeat. "Claude Code, Copilot and Codex All Got Hacked — Every Attacker Went for the Credential, Not the Model." VentureBeat, 2026.

[15] Proofpoint. "Access Granted: Phishing with Device Code Authorization for Account Takeover." Proofpoint Threat Insight, December 2025.

[16] Stytch. "Agent-to-Agent OAuth: A Guide for Secure AI Agent Connectivity with MCP." Stytch Blog, August 2025.

[17] Strata Identity. "2026 Guide to OAuth Token Exchange and Agentic AI." Strata Blog, 2026.

[18] Aembit. "AI Agent Identity: The Multi-Protocol Authentication Gap." Aembit Blog, 2026.

[19] GitGuardian. "AI Agents Authentication: How Autonomous Systems Prove Identity." GitGuardian Blog, April 2026.

[20] Goodwin Law. "Beyond the Perimeter: Securing OAuth Tokens and API Access to Thwart Modern Cyber Attackers." Goodwin LLP Insights, October 2025.

[21] CyberArk. "2025 Identity Security Landscape." CyberArk, 2025.

[22] Gartner. "Gartner Predicts 40% of Enterprise Apps Will Feature Task-Specific AI Agents by 2026, Up from Less Than 5% in 2025." Gartner Press Release, August 2025.

[23] IETF. "RFC 8628: OAuth 2.0 Device Authorization Grant." IETF, August 2019.