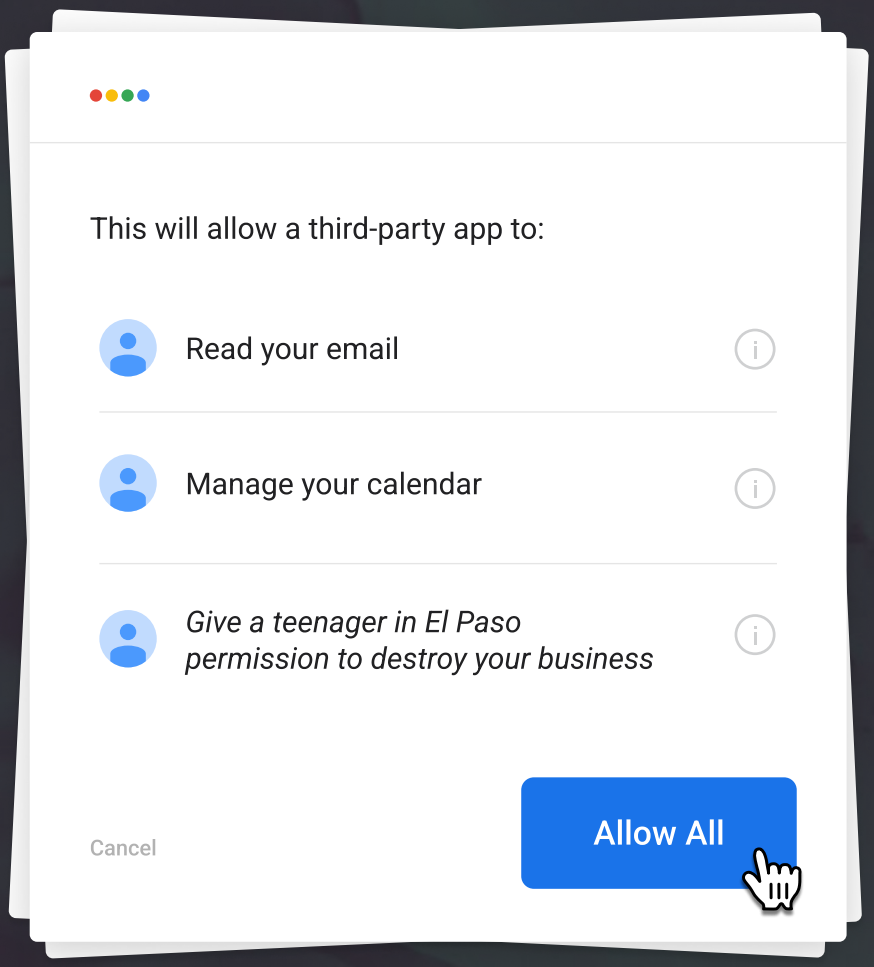
Material

OAuth & Google Workspace Risk Report



Seems fine.

Every OAuth grant can give third-party apps lasting access to your inbox, files, calendar, and identity data. Most teams don't know which apps have access, what they can reach, or who approved them.

What enterprise security teams can't see can still hurt them

OAuth has become the connective tissue of the modern cloud workspace. Employees link AI platforms, productivity tools, scheduling assistants, and corporate systems to their Google Workspace and Microsoft 365 accounts through a single authorization flow that is as easy to connect as it is difficult to manage. From coding assistants to calendar apps to document editors, everything connects back to the same account and carries the same implicit trust.

That ubiquity is what makes OAuth so difficult to govern. Block lists can't keep up with the pace of emerging tools, and allow lists create friction that slows teams down. Manual review at scale isn't feasible, and even where teams have the capacity to try, understanding what a connected application actually does, not just the permissions it requests, is often impossible from the outside.

AI EXPLOSION

91%

of AI apps appeared in the last 16 months

RESTRICTED SCOPES

24.5%

of apps hold restricted Google scopes

AI agents have made this problem acute. For years, employees connected tools that were largely passive: they read data, generated suggestions, and surfaced insights. The wave of AI agents now entering the workforce operates differently. These systems are built to act: their behavior is determined at execution rather than at the grant stage, not by accident but by design. They schedule meetings, send emails, create and modify files, execute multi-step workflows, often without a human approving each step. The mechanism that enables all of it is OAuth. Every agent that connects to your cloud workspace does so through an authorization someone granted in a few clicks, frequently without IT awareness, and almost never with a clear record of what the agent is permitted to do.

The result is an attack surface that most organizations have authorized but never fully mapped.

This report documents that gap through four findings from an analysis of 22,332 OAuth-connected applications across 21 enterprise Google Workspace environments. Each finding addresses a category of risk that does not appear in conventional application inventories: access that persists after use has stopped, connections that were never formally approved, and permissions that continue to operate with no active user behind them.

91%
of AI apps appeared in the last 16 months

47%
of apps unused in 90+ days

1064
apps have zero active users (tokens still live)

24.5%
of all apps hold restricted Google scopes

463
zombie-token apps with sensitive or restricted scopes

181
AI apps with sensitive or restricted scopes

These findings do not describe compromise. They describe the conditions that make compromise harder to detect and easier to sustain. The gap is not between what organizations allow and what is safe. It is between what organizations have authorized and what they are able to actively monitor.

91% of AI-connected apps appeared in the last 16 months

The dataset includes 356 unique applications classified in the AI and Automation category that hold Public Application status, meaning they are verifiable external tools rather than internal integrations. Of those, 325 were first observed in the environments analyzed on or after January 1, 2024: 91% of the AI application population appeared within a 16-month window.

356
AI and Automation Public apps total

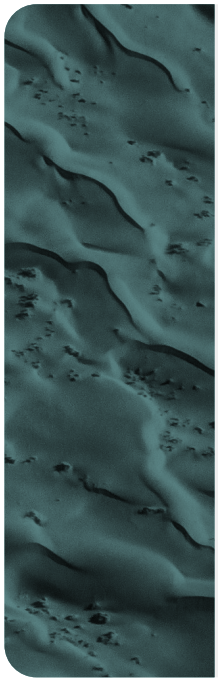
325
first seen since January 2024

>50%
AI apps hold sensitive or restricted scopes

9
months average age of AI-connected apps

42%
connected for over a year

149
apps connected 12+ months, no review



The app that wasn't Gamma

The name was familiar. Gamma is a widely used AI presentation tool, and "gamma.com.ai" looked close enough that plenty of users connected it without a second thought. The only problem? It wasn't Gamma. It was a separate app registered by a Chinese developer, packaged to resemble the legitimate service, and quietly requesting OAuth access to Google Workspace accounts across multiple client environments.

This is OAuth impersonation, and the attack surface it exploits is entirely human. No phishing email required, no MFA bypass, no malware. Just an employee recognizing a name, clicking Allow, and handing over access to email, Drive, or both. The attacker's work is largely done at that point.

The actual Gamma app (gamma.app) is unrelated to this developer or domain. But visual similarity is the mechanism. Most users aren't running a domain lookup at the moment they authorize an app; they see a name they recognize. That gap between the name and the domain is where the attack lives.

This rate of growth does not reflect a coordinated IT rollout. It reflects individual employees connecting AI tools on their own: developers integrating coding assistants, sales teams connecting AI drafting tools, HR teams linking AI scheduling assistants. Each authorization is made independently, often without IT visibility. The average AI-connected app in this dataset has been running for 9 months. Four in ten have been connected for over a year, spanning multiple budget cycles, team changes, and security reviews, with no formal record of approval.

What makes this pattern consequential is the access these tools require to function. AI assistants that help employees write emails need to read emails. AI tools that help employees manage their calendars need full calendar access. AI coding tools that work with documents need Drive access. Of the 181 AI applications holding sensitive or restricted scopes, most are doing exactly what they were connected to do. The problem is not the functionality. It is the absence of any record that the access was deliberately approved.

For AI agents, the stakes are higher. Agents are built to act autonomously within the permissions they hold. A passive tool that can access your mailbox can see your email. An agent with the same scope can read it, respond to it, and forward it, with no human approving each step. An OAuth grant to an agent is not access to data. It is operational authority. Exactly what that agent is doing with this operational authority is something more organizations cannot answer.

DID YOU KNOW...

...most AI tools connected to your Google Workspace were never formally approved? Someone clicked authorize, and that was it. No review. No owner on record. No process for what happens next.

Organizations that do not have a lightweight approval process for AI application connections are not preventing those connections from happening. They are making sure they happen without oversight.

> "OAuth is the path of least resistance for app login, which is exactly why it's so widely abused. Most users have no idea what they're agreeing to when an app asks for permissions. They click allow and move on. In the AI era, allowlisting slows people down, but it's still what most security teams prefer because at least it gives you some visibility. You're choosing between friction and blind spots."

FRANK WANG
Security Engineer @ Surge AI
and author of Frankly Speaking

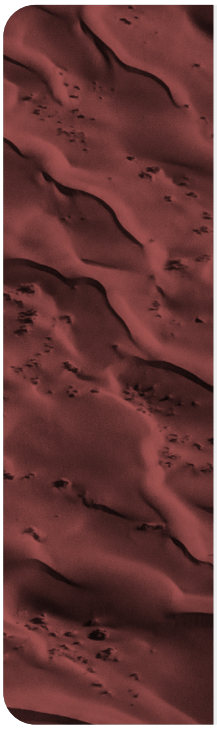
Nearly half of all OAuth apps are dormant. Their access is not.

Of the 22,332 unique applications identified across the environments analyzed, 10,545 (47.2%) have not recorded active usage in the past 90 days. A quarter of all applications, 5,752 in total, have not been used in 180 days or more. In every one of these cases, the OAuth authorization is still intact. The application retains whatever permissions it was originally granted.

47.2%
not used in 90+ days

25.8%
not used in 180+ days

22,332
total apps analyzed



Clockwise still has the keys

Clockwise was a well-regarded AI calendar assistant that was acquired by Salesforce and subsequently wound down. At some point the service went dark, active development stopped, and Clockwise ceased to exist as an independent product.

The OAuth grants didn't get the memo. Across several client environments in this assessment, Clockwise still appeared in connected app inventories with active grants, including read and write access to calendar and email. The vendor is gone. The security team maintaining it is gone.
The access remains.

This is the quiet risk that stale access creates. The threat isn't necessarily active exploitation; it's that the attack surface exists without anyone watching it. A decommissioned app doesn't have an incident response process anymore. It doesn't have someone patching vulnerabilities or responding to abuse reports. Regular access reviews surface this. Without them, old grants accumulate indefinitely.

OAuth tokens do not typically expire automatically when an employee stops using an application. They will expire only when the user manually revokes access, when an administrator removes the authorization, or when a token tries to refresh but the account behind it is no longer active. In practice, most tokens remain active indefinitely unless someone takes deliberate action to remove them.

This creates a class of risk that conventional security tooling rarely surfaces: live credentials granted to applications that no one in the organization is actively monitoring. If an application is compromised, if its vendor is acquired, or if its API behavior changes, the organization will not learn about it through normal usage signals. There are none.

DID YOU KNOW...

...apps do not clean themselves up when you stop using them? The access stays. The token stays. The risk stays. Only a deliberate revocation makes it go away.

The practical question for each dormant application is straightforward: does the business still need this connection? If the answer is no, the authorization should be revoked. If the answer is yes, it should be documented and assigned an owner. In most cases, neither has happened.

"When I get a new vendor request, part of my risk assessment is knowing exactly what access they're asking for. Ninety-nine percent of the time, the business requester has no idea. We do the risk assessment at the grant: read-only, approved, move on... but then four months later we find out it's actually full read/write. Wild West permissions. Material's automated approach is a night-and-day improvement."

Head of Security @ Cloud-Native Security Platform

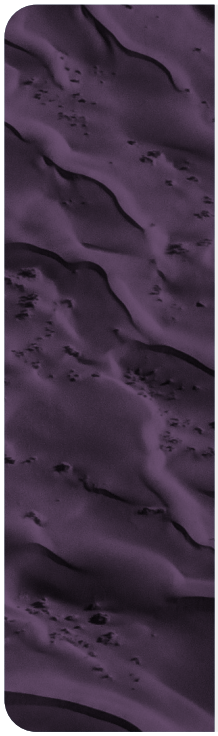
1,064 apps hold live tokens for users who are no longer there

Here's something that continually surprises even seasoned security practitioners we speak with: OAuth apps don't automatically lose access when an account is suspended. A subset of applications in the dataset presents a sharper version of this very dormancy problem: 1,064 unique applications show zero current active users alongside positive historical usage. These are applications that employees once connected and then stopped using, or applications connected to accounts that have since been suspended, whose OAuth tokens were never revoked.

1,064
apps with 0 active users, live tokens

463
zombie-token apps with sensitive or restricted scopes

43.5%
of zombie-token apps hold sensitive or restricted scopes



The movie app that fooled the scanners

NetMirror is a free movie streaming app distributed as an APK outside the Play Store. Material flagged it across multiple client environments based on its runtime behavior. When we investigated further, we found independent security research that explains why it's notable beyond being a sketchy sideloaded app: it was engineered specifically to evade automated analysis tools, hiding its malicious logic in a binary format most APK scanners don't inspect and going dormant whenever it detected a sandbox or emulator environment. Automated scans returned clean results by design.

On a real device, the app collected permanent hardware identifiers, intercepted credentials entered in its built-in browser, generated fraudulent ad clicks in hidden WebViews, and contained a mechanism to push a second-stage payload from attacker-controlled infrastructure. For any employee who had this on a device with access to corporate accounts, the exposure extended well past the personal.

The presence of an app like this in a professional environment starts as a shadow IT problem. The malware is secondary to the visibility gap that let it get there.

These are not dormant applications in the conventional sense. They are applications where the original authorizing user is no longer active, whether because the employee left the organization, changed roles, or simply stopped using the tool. In each case, the application continues to hold a valid credential that was issued by that user's account.

Of the 1,064 applications in this category, 463 hold permissions classified as sensitive or restricted, including full Gmail access, full Drive access, and in some cases broader Google Workspace administrative permissions. These are not low-stakes credentials. They are, in several instances, the most restricted permissions available, now attached to accounts that no active employee is monitoring.

An OAuth token issued by a former employee does not become invalid when that employee leaves. It remains valid until it is explicitly revoked.

DID YOU KNOW…

…OAuth apps can still reach data in a suspended account? Suspension cuts off the user. It does not cut off the apps. That step must be done separately, and most organizations skip it.

The security implication is not theoretical. A compromised application holding a valid token from a suspended account can exfiltrate data, send email, or modify files without triggering alerts based on user behavior, because the behavior appears consistent with a legitimate, if inactive, authorization.

> "One of the less obvious risks with OAuth is what happens after the fact. When you revoke a user's access or offboard a vendor, you're thinking about accounts and credentials — but OAuth grants operate on a different lifecycle. The approved app holds a token that can be refreshed indefinitely, which means processes can keep running long after the user has been suspended or the vendor relationship has ended. These are zombie connections — technically authorized, practically abandoned, and invisible to most of the controls we rely on."

CHAIM SANDERS
CISO @ Lyft

1 in 4 apps holds restricted Google scopes

Google's OAuth framework designates certain permission types as restricted: scopes that carry sufficient sensitivity to require enhanced review before an application can request them in production. Gmail and Drive are the most common. Across the 22,332 applications in this dataset, 5,461, nearly one in four, hold at least one active restricted scope type. This figure is not derived from a third-party risk model. It is based on Google's own classification of the permissions these applications have been granted.

5,461
apps with active restricted scope types

24.5%
of all 22,332 apps in the dataset

53.4%
of Public (Governable) Apps with sensitive or restricted scopes

The most common restricted scope types observed are Gmail and Drive, often appearing together. An application holding both has the ability to read all of an employee's email, send messages on their behalf, and access every file in their Google Drive, including files shared with them from other parts of the organization. This combination is the highest-consequence permission profile available in the Google Workspace OAuth framework.

This finding does not say that 5,461 applications are malicious or that their access is being abused. Many are well-known, widely used business tools that require these permissions to function. What it does say is that nearly one quarter of all applications in these environments hold permissions that Google itself designates as restricted enough to require special handling, and in most cases that special handling has not been applied in any documented way.

DID YOU KNOW…

…Google flags certain permissions as restricted because they carry too much power to go unreviewed? One in four apps in this dataset holds at least one. Most were never reviewed at all.

What this means for security leaders

The four findings in this brief share a common structure. In each case, the risk does not originate from an obviously bad decision. It originates from a reasonable one: connecting a useful tool, granting the access it needed, and never revisiting either. OAuth authorizations are persistent by design. The governance processes in most organizations are not.

The result is an attack surface that grows continuously, quietly, and in ways that do not appear in conventional security tooling. An application that was fully appropriate eighteen months ago may now be dormant, may have changed ownership, or may be connected to an account that no longer has an active employee behind it. None of those changes trigger alerts. None of them show up in access logs. None of them are visible without a deliberate effort to look.

A floor on visibility: 17,262 internal applications are not in this analysis

The four findings in this brief cover the 22,332 unique applications identified across the environments analyzed. Of those, 17,262 (77.3%) are classified as Internal Applications: custom-built scripts, internal automation, service accounts, and legacy pipelines. They are excluded from the comparative analysis because they lack the verifiable external identity required for benchmarking. That is a methodology choice, not a risk assessment.

Internal applications can hold the same Gmail, Drive, and administrative scopes as any third-party tool. They can be equally dormant and equally invisible to security tooling. The difference is that their risk can only be assessed by someone inside the organization who knows what they do. Addressing the public application findings in this brief without reviewing internal inventory resolves the measurable surface while leaving the larger unknown in place.

Visibility is the first problem. The second is that most organizations do not have a process for what to do once they have it. The following three actions address both.

Three actions that address the visibility gap

01

Connect OAuth revocation to employee offboarding.

Standard account suspended does not revoke third-party OAuth grants. Adding an explicit revocation step to the offboarding workflow closes the zombie token gap directly. This is the single highest-return governance action available.

02

Build a governed pathway for AI tool adoption before the next wave.

91% of AI apps in this dataset appeared in 16 months. A lightweight registration process, one that requires a business owner, a description of purpose, and acknowledgment of the permissions being granted, converts unmanaged individual connections into documented, reviewable assets.

03

Establish a dormancy threshold and act on it.

Define what dormant means for your organization. Ninety days is a reasonable starting point. Apply it systematically. For each application that crosses that threshold, confirm whether the access is still needed. If yes, document it. If no, revoke it.

"I went through this exact OAuth journey at a previous company, completely manually. It was painful: reviewing hundreds of apps, figuring out what's legitimate and what isn't. Material makes it so much easier than what I had to do before."

Security Engineer @ Digital Health Platform

For organizations who want to understand their OAuth risk, Material provides continuously-updated OAuth inventory and real-time detection and response for suspicious and malicious apps.