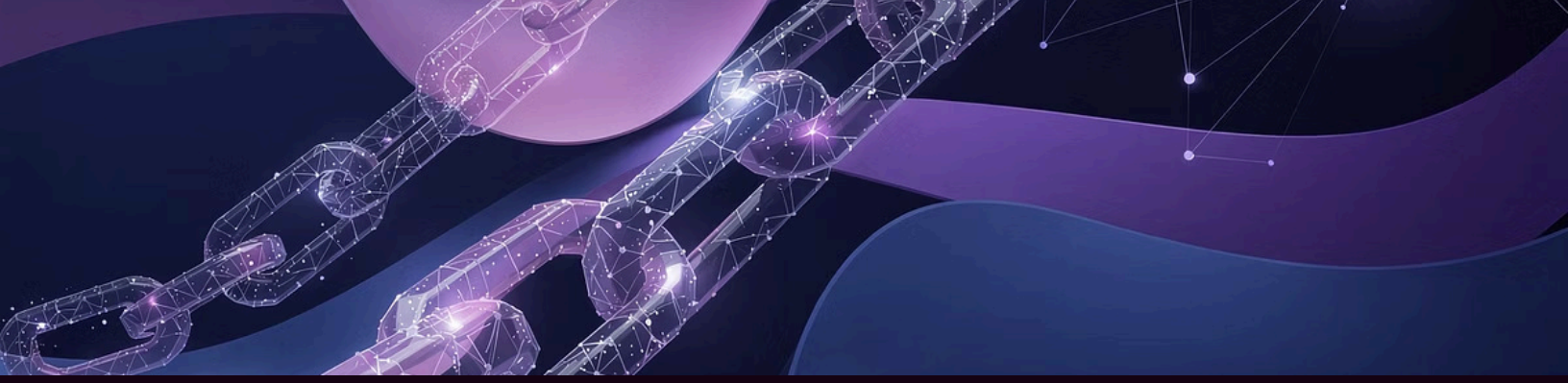

nudge

The OAuth & MCP Investigation Checklist

A practical guide for security and IT teams to spot risky grants, overly permissive scopes, and potential abuse—before attackers do.

Why this matters now

OAuth grants are the quiet backdoor of modern SaaS. When someone clicks "Sign in with Google" or approves a third-party app, they're handing another vendor a key to your data, often without IT or security in the loop. Most of those grants are low-risk. Some are not.

Recently, the picture got more complicated. Many remote **MCP (Model Context Protocol) servers**—the connectors that provide the data to grant AI agents and tools permission to read/write, search databases, or act on a user's behalf—use OAuth as their authorization layer. That means every AI agent that does is, underneath, another OAuth grant for you to track. A single MCP connection can give an AI agent broad read/write access across email, files, code, CRM, and more.

Recent incidents (the **Vercel OAuth compromise** being the latest high-profile reminder) show how quickly a single overly permissive grant can turn into a supply-chain problem. This checklist gives you a repeatable way to investigate any OAuth grant—whether it's a human-approved SaaS app or an MCP connection—before it becomes an incident.

When to run this checklist

Run through this any time:

New OAuth grant appears

A new OAuth grant appears in your environment (employee-approved, admin-approved, or auto-discovered).

MCP server shows up

An MCP server shows up as a connected app, especially one requesting broad scopes.

Vendor breach disclosed

A vendor you rely on discloses an OAuth-related breach. Revisit grants tied to that vendor.

Quarterly or annual review

You're doing a routine third-party access review on a quarterly or annual basis.

Something looks weird

Someone flags an app as suspicious. Trust the instinct, then work the list.

THE FOUR PILLARS

The four pillars of OAuth investigation

Every OAuth grant can be evaluated along the same four dimensions, whether it's a SaaS plugin, an MCP server, or anything else requesting access to your data. Work each one in order.

1

1. Scopes & Permissions

What the app is actually allowed to do with your data

2

2. App Registration

Who really owns this integration—the paper trail

3

3. Vendor Trust

The company behind the app: real, vetted, and accountable

4

4. Popularity & Usage

Who granted it, when, and is it still actively used

PILLAR 1

Scopes and permissions

Scopes are the *what*: what the app is actually allowed to do with your data. This is the single highest-signal check.

1. **List every granted scope**

   Not just what was requested—what was actually approved.

2. **Ask: does the app's purpose justify this access?**

   A note-taking bot does not need `mail.send`. A calendar assistant does not need full Drive read/write.

3. **Flag write, delete, admin, or full-access scopes**

   These deserve extra scrutiny. Flag any scope that crosses data boundaries (e.g., access to both email and files, or identity and billing).

4. **For MCP servers: expect broad scopes by design**

   MCP connectors often need wide read access to be useful to an AI agent. Decide whether that breadth is acceptable for *this* server from *this* vendor.

5. **Check for scope drift**

   Has this app quietly gained new scopes since it was first approved? Updated scopes mean a re-review.

🚩 **Red flag:** The granted scopes are broader than what the app's public documentation says it needs. Either the app is overreaching or the docs are out of date. Both are worth a vendor conversation.

PILLAR 2

App registration details

The registration metadata is your paper trail. It tells you who really owns this integration.

1. **Pull the publisher email / publisher domain**

   Does it match the vendor's real domain? A Salesforce integration published from a `gmail.com` address is a red flag.

2. **Check the reply / redirect URL(s)**

   Do they point to the vendor's production domain, or to something that looks like a staging server, a personal project, or a lookalike domain?

3. **Verify publisher verification status**

   E.g., Microsoft Publisher Verified, Google verified. Unverified isn't automatically bad, but it raises the bar for other signals.

4. **Look at when the app was registered**

   Brand-new registrations impersonating well-known brands are a classic phishing pattern.

5. **For MCP servers: confirm the server URL**

   Confirm it matches the vendor's real domain, and that the OAuth client is registered to the vendor—not an individual developer or a generic cloud host.

🚩 **Red flag:** The publisher email, redirect URL, and app name don't tell a consistent story. If "Acme Corp" the app is registered to a Hotmail address pointing at a Heroku URL, you have three different stories and one very likely problem.

PILLAR 3

Vendor trust signals

Zoom out from the app itself and look at the company behind it.

1 **Is there a real company?**

A real website, real people, and a real security page. Look for a privacy policy and data processing agreement you'd actually sign.

2 **Check for security certifications**

SOC 2, ISO 27001—and can you verify them? Search for the vendor name plus "breach," "vulnerability," "CVE," "incident." Anything recent?

3 **For MCP vendors: weight other signals more heavily**

The space is new and moving fast. Expect fewer certifications. Focus on company maturity, funding, team transparency, and public security posture.

4 **Check for downstream dependencies**

An OAuth grant to Vendor A might effectively give Vendor B access too, if Vendor A passes credentials or data downstream.

🚩 **Red flag:** You can't find a real business behind the app. No LinkedIn presence, no real support channel, a privacy policy that's a generic template. Don't hand your data to a ghost.

PILLAR 4

App popularity and usage

Popularity isn't proof of safety, but combined with the other signals, it's useful context.

1. **How many users have granted this app access?**

   A single user granting an obscure app → investigate. Company-wide adoption of a well-known app → lower bar.

2. **Is this a known app in the broader ecosystem?**

   Look for an established marketplace listing, thousands of other customers, and documented use cases.

3. **Which users granted it, and when?**

   Cluster approvals in a short window from unrelated users can indicate a phishing campaign.

4. **Is the app actively used, or stale?**

   Stale grants are free keys sitting in the lock. Revoke unused grants with broad scopes—legitimate users will notice and ask for access back.

5. **For MCP servers: one user = outsized access**

   A single power user connecting an MCP server to their AI assistant may represent outsized access, because one AI agent can act across every scope the MCP server has been granted.

🚩 **Red flag:** An app that nobody is actively using but still holds broad scopes. Revoke first, ask questions later. Legitimate users will notice and ask for it back.

SPECIAL ATTENTION

MCP server connections

MCP servers deserve a dedicated look because they change the shape of OAuth risk in a few important ways:

They're data highways, not point tools

A single MCP server can expose many tools at once (email + calendar + files + code). The blast radius of a compromised or misconfigured MCP connection is far wider than a typical single-purpose SaaS integration.

The "user" may be an AI agent

That agent doesn't pause to question a suspicious request. Whatever scopes are provisioned, assume they will be exercised often—and in combinations a human would never choose.

Auth patterns are mixed

OAuth connections are discoverable through standard app integrations, but API keys and credentials configured locally may leave no trace. Some locally-run MCP servers require no authentication at all. Don't assume absence of an OAuth grant means absence of a connection.

Broad scopes by design

This isn't inherently malicious—MCP servers are meant to be general-purpose—but it means "the scope is broad" alone isn't disqualifying. Pair scope review with strong vendor-trust review.

They're often shadow IT

Developers and power users hook up MCP servers without a ticket. Your first investigation may double as your first inventory.

Extra checks for MCP connections

1 **Remote vs. local MCP server**

Identify whether the connection is to a **remote MCP server** (hosted by a vendor) or a **local MCP server** (running on an employee's machine). Both carry risk; the mitigations differ.

2 **Confirm the MCP server's identity**

Is this the official server from the vendor, or a community re-implementation?

3 **Map the server to its AI client(s)**

Identify which agents are using it (Claude, ChatGPT, Cursor, an in-house agent, etc.). One MCP server connection can serve multiple agents.

4 **Check for corresponding API keys**

If the MCP server uses keys instead of OAuth, your OAuth inventory won't catch it. Your secret-scanning and SaaS-discovery coverage needs to.

5 **Sanctioned AI use case or shadow AI?**

Decide whether this MCP connection belongs to an approved use case. Document the decision either way.

6 **Review exposed tools, not just scopes**

Scopes define what access is authorized; tools define what actions an agent can actually take. A server with a broad tool surface warrants closer vendor scrutiny even if the individual scopes look benign.

A quick decision matrix

Use this matrix to triage any OAuth grant or MCP connection. When signals conflict, weight **scopes** and **vendor trust** most heavily. Broad access from a vendor you can't vet is the pattern that hurts.

Signal	✅ Lower risk	🔍 Investigate further	🚩 Revoke
Scopes	Read-only, narrowly relevant	Write access to one system	Admin, multi-system, or unexpected write access
Publisher	Matches vendor's real domain, verified	Unverified but otherwise consistent	Mismatched, generic, or freshly registered
Vendor trust	Established vendor, security page, certifications	Smaller vendor, thin public footprint	No real business behind the app
Popularity	Widely adopted inside and outside your org	Used by a few, new to your org	Unused for months, or sole adopter holds broad scopes
MCP specifics	Known vendor's official MCP server, scoped narrowly	Official server with broad scopes, sanctioned AI use case	Unofficial server, unknown vendor, or shadow AI use

Don't stop at day one

Investigation isn't a one-time gate. OAuth grants and MCP connections change over time, often silently.

Set a review cadence

Quarterly at minimum, monthly for high-risk scopes. Treat reviews as a standing operational task.

Get alerted on new connections

Especially for scopes you've flagged as sensitive. Also alert when an existing app gains new scopes.

Revoke on offboarding

Revoke grants for offboarded employees as part of your offboarding checklist—not three months later.

Keep a running log of decisions

Approved, restricted, revoked, and why. Future-you will thank present-you.

Make OAuth and MCP investigations routine

The Vercel incident and others like it are reminders, not outliers. As MCP adoption accelerates, the volume of OAuth connections in the average environment is only going up—and so is the blast radius of each one. The teams that handle this well aren't heroically investigating one app at a time; they have a repeatable playbook, a live inventory, and alerts when something changes.

This checklist is that playbook, on one page. Print it, paste it into your runbook, or plug it into your existing third-party review process. Next time a new grant shows up—SaaS plugin, MCP server, or whatever comes after—you'll know exactly what to do.

Pro tip: When signals conflict across the four pillars, always weight **scopes** and **vendor trust** most heavily. Broad access from a vendor you can't vet is the pattern that hurts.

How Nudge Security helps

Running this checklist once is useful. Running it automatically and continuously—against every new grant that shows up in your environment—is better.

Discover OAuth grants & MCP servers

Nudge Security automatically surfaces OAuth grants and remote MCP servers across your entire SaaS stack as they appear.

Surface high-risk connections

Get scopes and publisher details you'd otherwise have to chase down by hand, including flags for overly permissive grants.

Alert on changes

Get notified the moment something changes—a new app, a scope update, a new MCP server connection—so nothing slips through.

You bring the judgment; we bring the full picture.

Start a free trial | See how it works