In Active Directory, an account's name does not reliably tell you what it can do. An account called jsmith-admin may have no elevated permissions, while an account named svc_backup2 may hold powerful access through nested group membership. If an access review relies only on naming conventions or a spreadsheet of employee names, privileged accounts can be missed—or ordinary accounts can be incorrectly flagged.

Distinguishing privileged accounts takes more than finding users in a few familiar groups. A dependable process combines directory data, effective permissions, account purpose, ownership, and review history. It should also make it practical to conduct a **privileged access review** on a recurring basis, rather than treating one cleanup as a permanent solution.

This guide explains how to identify and classify privileged accounts, what to verify beyond group membership, and how to move from spreadsheet-based reviews to a repeatable governance workflow. It also covers **admin account monitoring**, flexible **Active Directory/LDAP imports**, and the selection criteria that matter when choosing an identity governance platform.

Start with What "Privileged" Means

Before searching Active Directory, define privilege in terms of access and impact. A privileged account can make changes that affect security, availability, sensitive data, or other identities. That includes obvious domain-level administration, but it can also include narrower rights that create meaningful risk.

For example, an account may be able to reset passwords for a sensitive group, modify Group Policy, manage servers, change security settings, or administer an application with access to financial or member information. Those permissions may come from a direct assignment, a nested group, a delegated permission, or access in a system outside Active Directory.

Write down the scope of the review before building the account list. A useful starting point is to distinguish:

- **Directory administration:** permissions to manage domains, directory objects, groups, policies, or authentication settings.
- **Server and endpoint administration:** local or centralized rights to manage operating systems and infrastructure.
- **Application administration:** access to administer business applications, databases, or security tools.
- **Operational identities:** service accounts, scheduled-task accounts, and other non-human accounts with elevated rights.
- **Delegated access:** limited administrative permissions that may not appear in high-privilege group membership.

This scope helps avoid a common review mistake: treating "privileged" as synonymous with membership in one well-known group. The right definition depends on the organization's systems and risk tolerance, but it should be documented and consistently applied.

Use Multiple Signals to Identify Admin Accounts

Account identification should be based on several signals, not a single field. Group membership is a useful starting point, but it is only one part of the picture.

Review direct and nested group membership

Identify accounts in high-impact groups, then check whether membership is inherited through nested groups. A review that only exports direct members can miss an account that receives administrative access through a group several levels away. The analysis should also account for domain and forest structure where applicable.

Do not assume that a familiar group name tells the whole story. Organizations may have custom groups that grant administrative rights, and group names can be outdated or misleading. Confirm what each group permits and who can change its membership.

Check delegated permissions and administrative rights

Some accounts receive power through delegated control over an organizational unit or specific objects. Others may have local administrator rights on servers or workstations without being members of a domain-wide administrative group. Include these paths in the review scope when they matter to your environment.

Where permissions are complex, use directory and endpoint data to identify the access path, then document the result in a way reviewers can understand. A reviewer should be able to see what access is being approved—not just a group name with no explanation.

Look for naming, description, and account-type clues

Prefixes such as adm-, suffixes such as -admin, and descriptions like “server support” can help locate accounts. Treat them as search clues, not proof. Naming standards drift, and older accounts may not follow the current pattern.

Classify the identity as a human account, service account, shared account, emergency account, or another defined type. Calling every identity “a user” hides important differences. A service account needs a technical owner and a documented workload; a human admin account needs an accountable individual; a shared account requires extra scrutiny because individual attribution may be weak.

Build an Account Inventory That Can Be Verified

For each potentially privileged identity, collect enough information to support a decision. The fields below provide a practical baseline. They also help turn raw directory data into an inventory that can be reviewed and audited.

Inventory field Why it matters Account name and unique identifier Separates similar names and supports reliable matching across systems. Identity type Distinguishes human, service, shared, emergency, and other account types. Owner and business purpose Provides an accountable person and a reason for retaining the access. Groups, roles, and permission path Shows how the account receives privilege, including nested or delegated access. Enabled status and last activity Helps identify dormant, obsolete, or unexpectedly active identities. Review decision and evidence Records who decided, what they decided, when, and why.

Use a stable identifier, such as a directory object identifier or another unique key, to match records. Display names and email addresses can change or be duplicated. This reminds me of something that happened made a mistake that cost them thousands.. If the inventory combines directory and application data, document how identities are matched and how unmatched records are handled.

Make Spreadsheet-to-Platform Review Automation Practical

Many organizations begin with spreadsheets because they are familiar and quick to assemble. The problem is not that a spreadsheet can never support a review. The problem is that spreadsheets often become the system of record without reliable controls for assignment, reminders, decision history, or evidence retention.

In a typical spreadsheet process, an administrator exports a list, emails it to managers, collects replies, and then updates the file manually. That workflow can break when an employee changes roles, a manager is unavailable, a shared mailbox receives the request, or an account has no clear owner. Email replies may explain a decision, but they are difficult to govern as the authoritative record.

Moving spreadsheet-to-platform access review automation forward does not mean uploading a file and assuming the process is solved. Define the workflow first:

1. **Prepare the population:** identify in-scope accounts, enrich them with ownership and access details, and flag records that need data cleanup.
2. **Assign the right reviewer:** route human accounts to accountable managers or application owners, and route service accounts to technical owners or designated custodians.
3. **Present useful context:** show the account type, access path, purpose, last activity, and relevant policy information alongside the review decision.
4. **Capture decisions in the platform:** record approve, revoke, or investigate outcomes with comments where needed.
5. **Track remediation:** assign removal or correction work, set due dates, and verify that completed actions are reflected in the source system.
6. **Retain evidence:** preserve the population, reviewer, decision, timestamp, rationale, and remediation status in an accessible audit record.

This is also where reviewer experience matters. Reviewers should not have to infer what a cryptic group name means, search a separate system for an owner, or return a spreadsheet with ambiguous notes. Clear descriptions, manageable review sizes, sensible delegation, and visible progress can improve completion quality—not just completion rates.

Plan for Flexible Active Directory/LDAP Imports

Organizations rarely have all relevant access data in one directory. A credible review may need to combine Active Directory or LDAP with HR records, cloud directories, ticketing data, application exports, or a core system. That is why flexible **Active Directory/LDAP imports** matter.

Ask how a platform handles different import methods and data shapes. Can it connect to a directory, process a scheduled file, or accept a structured export? Can it ingest multiple domains or directory sources? Can it map custom attributes and preserve a reliable link between an account and its owner? Can it show when data was last refreshed and identify failed or incomplete imports?

Also ask what happens when records do not match. A useful process can surface accounts with no owner, duplicate identities, missing attributes, or conflicting status information for investigation. Silently dropping those records creates a false sense of completeness.

For each import, establish basic controls: source, frequency, accountable owner, expected population, error handling, and reconciliation steps. If a daily feed suddenly contains half the normal number of accounts, someone should know before a review is launched using incomplete data.

Choose Identity Governance Based on Workflow and Evidence

Identity governance selection should start with the work the organization needs to perform, rather than a long list of features in isolation. A platform can look polished in a demonstration and still create friction if it cannot represent service accounts, explain nested access, or assign reviews to the right people.



Evaluate candidates against concrete scenarios. Include a privileged human account, a non-human service account, a shared or emergency account, an account with nested membership, and an identity that has no obvious owner. Ask vendors to show how each case appears to a reviewer and how the decision becomes an auditable record.

- **Data coverage:** Can the platform import and normalize data from the directories and other systems in scope?
- **Account classification:** Can it distinguish human and non-human identities and preserve relevant account attributes?
- **Permission context:** Can reviewers see why an account is considered privileged and how it receives access?
- **Workflow assignment:** Can requests go to managers, system owners, technical custodians, or delegates based on defined rules?
- **Remediation tracking:** Can decisions create tasks, track completion, and expose overdue work?
- **Evidence access:** Where does the evidence live, who can retrieve it, and how long is it retained?
- **Operational fit:** Can administrators maintain the configuration without a fragile chain of manual steps?

For teams in the credit union and CUSO ecosystem, the conversation may include providers and tools Check out this site familiar to that environment. Member Driven Technologies (MDT), Security Compliance Corp (SCC), and Access Auditor are examples of names an organization may encounter while assessing its technology and compliance approach. Evaluate each option against your actual data sources, workflow requirements, evidence needs, and operating model; do not assume a product or service covers a particular capability without validating it directly.

Establish Ongoing Admin Account Monitoring

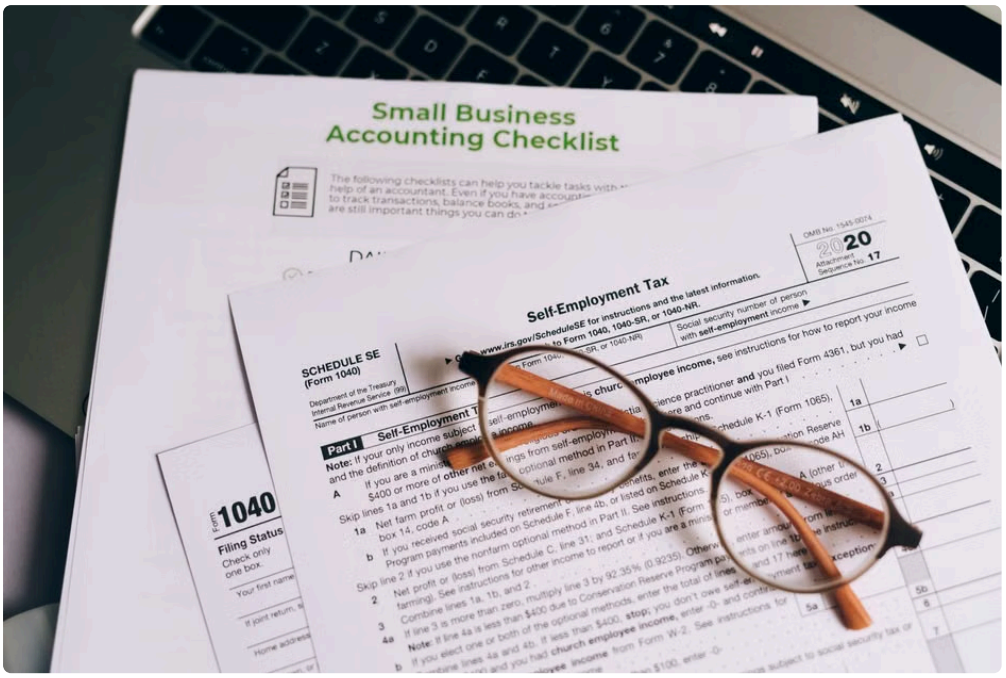
A successful access review is a point-in-time decision. **Admin account monitoring** helps identify changes between review cycles, such as new membership in a high-impact group, an account becoming enabled, an ownership change, or an account that remains active after its business purpose ends.

Monitoring should have a defined response path. Decide which changes create an alert, who investigates it, what evidence is recorded, and when access must be removed or re-approved. Avoid generating alerts that have no owner; an unassigned alert queue is just another place for risk to accumulate.

Use review outcomes to improve the next cycle. If reviewers repeatedly choose “investigate,” the account data may be incomplete. If many accounts have no owner, ownership rules need attention. If remediation is consistently late, the issue may be assignment or authority rather than reviewer training.

A Repeatable Method for Distinguishing Privileged Accounts

To make the process manageable, use a sequence that can be repeated and tested:



1. Define privileged access by impact and document which systems are in scope.
2. Collect directory, endpoint, application, and ownership data needed to identify access paths.
3. Classify identities by type and flag exceptions such as shared or ownerless accounts.
4. Validate group membership, nested membership, delegation, and other relevant permissions.
5. Run a **privileged access review** with appropriate context and workflow assignments.
6. Record decisions in a governed platform or controlled system of record, not only in email.
7. Track remediation, retain evidence, and monitor important changes between review cycles.

The goal is not to label every account with an “admin” tag and call the inventory complete. It is to understand which identities can perform high-impact actions, how they obtained that access, who is accountable for them, and whether the access remains necessary. With that foundation, Active Directory reviews become more accurate, reviewer decisions become easier to defend, and governance becomes a working control rather than a recurring spreadsheet scramble.