When an employee has one account for everyday work and another for administrative tasks, an access review can easily miss the point. A reviewer may confirm that the employee still works for the organization while overlooking a powerful account that can reset passwords, change configurations, or access sensitive data. The reverse can happen, too: a legitimate administrative account may be mistaken for an unnecessary duplicate and removed without checking its purpose.

Separating admin accounts from standard user access is therefore more than a naming exercise. It requires a reliable inventory, clear ownership, risk-based review criteria, and evidence showing what reviewers actually approved. For credit unions and other regulated organizations, a disciplined **privileged access review** supports stronger access-control testing and helps auditors understand whether elevated access is appropriate, limited, and monitored.

Why separate administrative and standard access?

Standard accounts are generally used for routine activities such as email, document access, and business applications. Administrative accounts have elevated permissions that can affect systems, data, security settings, or other identities. Keeping these roles separate can reduce the chance that routine activity is performed with more privilege than necessary.

For an audit, the separation also makes the review more meaningful. Instead of asking only, “Does this person need access?”, reviewers can ask distinct questions:

- Does the person still need the standard account for their job?
- Is the administrative account still required, and for which specific duties?
- Are its permissions limited to the systems and actions the person supports?
- Is the account individually attributable to an authorized person?
- Is there evidence of approval, use, and periodic review?

These questions matter in environments with Active Directory, core banking platforms such as Symitar, infrastructure tools, and third-party services. The account inventory may include more than named employee accounts: service accounts, emergency or break-glass accounts, shared mailboxes, vendor identities, and administrative clones can all affect the review. Treating every identity as “a user” obscures important differences in ownership and risk.

Build an inventory that distinguishes people from accounts

Start by identifying the sources that contain identity and permission data. Depending on the organization, those sources could include a directory, application exports, a core system, a privileged access tool, cloud consoles, and spreadsheets maintained by system owners. Record what each source can and cannot tell you. A directory may show group membership, for example, but not the business reason a person needs access to a particular application.

Then create a useful relationship between the person, the account, and the access being reviewed. A person can have multiple accounts, and an account may not represent a person at all. At minimum, retain fields such as:

- Account identifier and account type, such as standard, administrative, service, vendor, or emergency.
- Named owner or accountable business owner, where applicable.

- System, environment, and access source.
- Role, group, entitlement, or permission being reviewed.
- Account status, last activity data if available, and relevant dates.
- Business purpose, approval reference, and reviewer decision.

Use consistent labels, but do not assume that labels alone prove an account is privileged. Names such as “admin,” “svc,” or “test” can help identify records, yet naming conventions are often incomplete or outdated. Validate elevated access against actual group membership, roles, permissions, or application configuration. This is a core part of defensible **access-control testing**: test the control that grants access, not just the description attached to an account.

Define the review rules before launching a campaign

A good review begins with a clear scope. Decide which systems, account types, and permissions are included, who is responsible for reviewing them, and how exceptions will be handled. If administrative access is scattered across multiple exports, establish how records will be reconciled and how duplicates will be treated.

Separate the review decisions as well. A reviewer might need to confirm the account owner, approve or revoke a permission, attest to a business purpose, or flag an account for investigation. A single “approve” button can be too vague when the reviewer is being asked to validate several different facts.

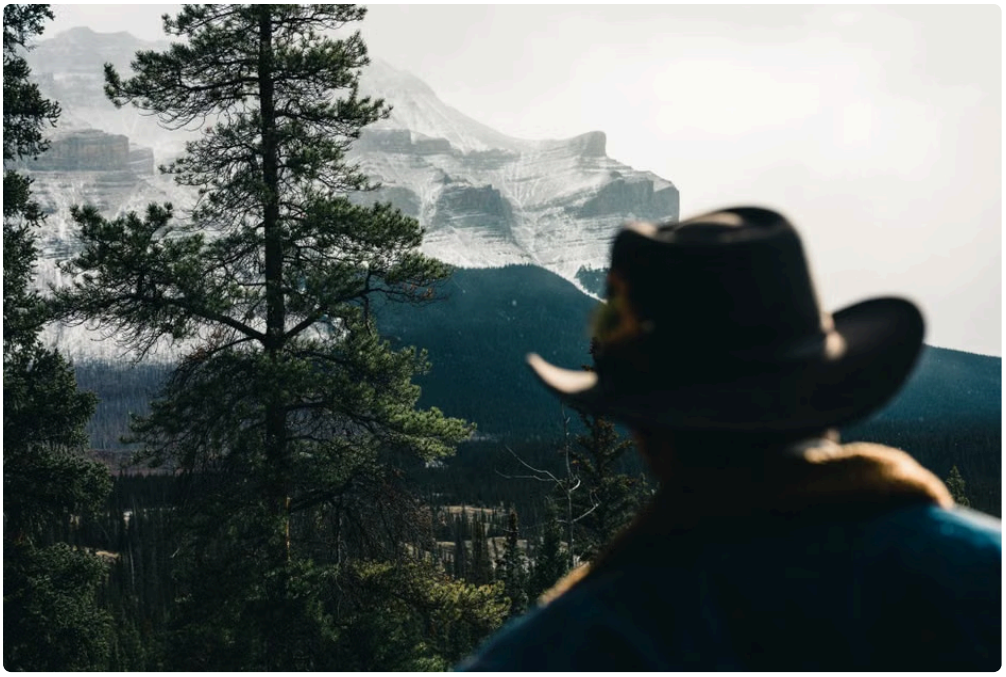
Set risk-based review expectations

Not every account needs the same level of scrutiny. A named administrator with broad directory permissions should receive more focused attention than a standard account with limited application access. Service accounts need a different review approach again: the reviewer should confirm the supported process, accountable owner, required permissions, and whether credentials or usage are managed appropriately.

For administrative accounts, make the review prompt specific. Ask the reviewer to verify the task the account supports, the system or environment in scope, and whether the level of privilege remains necessary. Where an account is not linked to a current owner, route it for investigation rather than quietly assuming the manager of a similarly named employee is responsible.

Move from spreadsheets to platform-based review automation

Spreadsheets are a common starting point for access reviews. They can help teams reconcile exports and establish a baseline, especially when data is coming from systems that do not integrate directly. But spreadsheet reviews can become difficult to control as the number of applications, identities, and reviewers grows. Versions proliferate, decisions arrive by email, and it becomes harder to show which file was authoritative or whether every item received a decision.



Moving from spreadsheet-based reviews to a platform does not make the process effective by itself. The workflow still needs clear ownership, useful review context, controlled decision options, and a reliable record of outcomes. Automation should reduce manual handling while preserving the evidence thesiliconreview.com needed to understand the review later.

A well-designed process typically follows these steps:

1. **Collect:** Import or connect to identity and entitlement data from the systems in scope.
2. **Normalize:** Map source fields into consistent account types, owners, systems, roles, and risk categories.
3. **Scope:** Select the accounts and permissions to review, including separate treatment for privileged and non-human identities.
4. **Assign:** Route each review item to a person with enough knowledge and authority to make the decision.
5. **Decide:** Capture approve, revoke, investigate, or other defined outcomes with comments when needed.
6. **Remediate:** Assign removal or correction tasks, then record completion and any exceptions.
7. **Retain evidence:** Preserve the source data, review assignments, decisions, timestamps, and remediation status.

Keep the system of record clear. Email may notify a reviewer that an item is waiting, but an email reply should not be the only evidence of the decision. Likewise, screenshots are usually a poor substitute when application or platform logs can show the decision, the actor, and the time directly. Before buying a tool, ask: **Where does the evidence live?** Confirm how records are retained, exported, and made available to audit.

Choose identity governance tools around the real workflow

Identity governance selection should begin with the organization's data and review requirements, not a feature checklist in isolation. A solution that works well for a clean directory feed may struggle when the review depends on different formats from a core system, a vendor portal, and a manually maintained application list.

Evaluate whether a platform can support the following:

- **Flexible data imports:** Can it accept structured files from many systems, handle changing columns, and map source fields without extensive rework?
- **Account and entitlement distinction:** Can it distinguish a person from multiple accounts, and identify service, vendor, shared, emergency, or administrative identities?
- **Useful review context:** Can reviewers see the system, permission, owner, business purpose, and relevant risk information without consulting several separate files?
- **Workflow assignment:** Can items be routed by application owner, manager, technical owner, or another appropriate reviewer? Can assignments be reassigned with a record of the change?
- **Reviewer experience:** Is it straightforward to review a focused queue, understand each decision, and document exceptions? Can reviewers handle volume without relying on email threads?
- **Remediation tracking:** Can revocations and corrections be assigned and followed through to completion?
- **Evidence and reporting:** Can the organization retrieve source data, assignments, decisions, timestamps, comments, and unresolved exceptions for audit?

Ask vendors to demonstrate a realistic scenario rather than a polished sample workflow. For instance, provide a mixed file containing employee accounts, an administrative clone, a service account with no obvious person-owner, and a shared mailbox with delegated access. See how the platform handles field mapping, reviewer routing, and exceptions. Companies such as Member Driven Technologies (MDT), Security Compliance Corp (SCC), and Access Auditor may come up in a financial institution's technology or compliance discussions. Evaluate any provider or product against your actual systems, import formats, workflow needs, and evidence requirements rather than assuming that a name or category guarantees a particular capability.

Design assignments for reviewers who know the access

Manager review alone is not always sufficient. A manager may know whether an employee still works on a team but not whether that employee needs a particular directory group or Symitar role. Conversely, a technical administrator may understand what a permission does but lack authority to confirm the business need. The workflow should send the right question to the right reviewer—or use separate business and technical approvals when both perspectives are needed.

Reviewer experience affects review quality. Keep assignments focused, use plain-language descriptions, explain what an approval means, and provide a clear path for uncertain items. Avoid presenting a long list of cryptic group names without system or business context. For privileged accounts, make it easy to identify the associated standard account without treating the two as interchangeable.



Escalation rules should also be explicit. Define what happens when a reviewer is overdue, unavailable, or unable to attest. An item should not become approved simply because nobody responded. Record delegation and reassignment in the workflow, and retain the identity of the person who made the final decision.

Test the control and preserve the audit trail

After the review, test whether decisions led to the intended outcomes. If an administrative account was revoked, verify that the permission was removed from the source system or that a documented exception explains why it remains. If an account was retained, confirm that the approval is tied to the relevant account and access—not merely to the employee in general.

For access-control testing, select samples across different account types and systems. Trace each item from source data through assignment, reviewer decision, and remediation. Check for gaps such as unassigned items, missing owners, approvals made by inappropriate reviewers, or revocations that were never completed. Keep a record of the population and selection method so the test can be repeated and understood.

Evidence should tell a coherent story: what access existed, who reviewed it, what they decided, when they decided, and what happened afterward. Retain the source exports or connector results, field mappings, campaign scope, workflow history, decision details, remediation records, and documented exceptions. Where the underlying system has useful logs, retain or reference those logs rather than relying solely on screenshots.

A practical standard for separating admin accounts

A reliable process does not require every system to use the same technology. It does require consistent treatment of identity, ownership, risk, and evidence. Start by distinguishing people from accounts, and standard access from elevated access. Make the purpose of each administrative account visible to

reviewers. Assign decisions to people who understand the business need and the technical permission. Then verify that approved changes were completed.

When the workflow is clear, spreadsheet-to-platform access review automation can reduce repetitive coordination without hiding the underlying control. The result is a more useful **privileged access review**, stronger **admin account monitoring**, and more defensible **access-control testing**—with evidence an auditor can follow from the original entitlement to the final outcome.