Simeio

Discover Privileged Accounts From Your Application

A practitioner's path to closing the gap between privileged account discovery and onboarding.

Executive Summary

As organizations add cloud platforms, SaaS tools, and multiple identity providers (IdPs), privileged accounts accumulate faster than people-centric identity and access management (IAM) programs can track. The result is privilege creep, dormant and orphaned account growth, overprivileged access explosion, and hardcoded access hiding in places routine reviews skip.

Closing the gap starts with discovery that matches the application: each application type carries privileged accounts at different layers, and knowing the type tells you where to look. What discovery surfaces then has to be onboarded, and that is where identity orchestration earns its place, turning scan results into risk-scored, onboarded, governed applications.

Table of Contents

Executive Summary 01

The Problem: Privileged Accounts Are Outpacing Discovery and Onboarding 03

Why Do Privileged Accounts Accumulate in Applications? 04

How Application Type Determines Discovery Depth 05
- 1. SaaS 05
- 2. Cloud-hosted legacy applications 05
- 3. Modern applications (containers and serverless) 05
- 4. Datacenter-hosted legacy applications 05
- 5. Homegrown legacy applications 05

Where Do Privileged Accounts Hide in Organizations? 06
- 1. Privileged Personal Accounts 06
- 2. Privileged Shared Accounts 06
- 3. Privileged Technical and Service Accounts 06
- 4. Orphaned Accounts 06

How Should You Discover Privileged Accounts? 07

Protect Privileged Accounts at Scale with Identity Orchestration 08

Move From Exposure to Durable Control 09

FAQs 10

Simeio



The Problem:
Privileged Accounts Are Outpacing Discovery and Onboarding

Every application an organization adds brings its own privileged identities and credentials. Many of these accounts carry broad standing access because that's how the application was set up, and they keep it because no one has found them and brought them under management.

Standing access has its benefits. It reduces operational friction for frequently accessed resources, lets administrators avoid repeated approval cycles that stunt productivity, and speeds incident response when access approvals could delay critical remediation. But those benefits only hold when privileged access is known, owned, and governed. In most enterprises, it isn't.

Organizations deploy resources in the cloud, on-premises, and in hybrid environments, with each account storing and granting access and permissions differently. Security teams may have privileged access management (PAM) tools in place, but the gap between discovering privileged accounts and onboarding them into those tools keeps widening. Accounts in that gap remain unmonitored and unmanaged, creating privilege creep that leaves organizations vulnerable to intrusions.

Most organizations believe they have done discovery because they scanned a directory and filed the results. That scan catches one layer. The accounts inside applications and on the systems beneath them never show up, so the inventory looks complete while it isn't. The real exposure sits in everything the scan never touched.

Why Do Privileged Accounts Accumulate in Applications?

Privileged accounts accumulate because every environment an organization adds, on-prem or cloud, brings identities and access that no single system consolidates. The more resources organizations use in different areas, the more identities and access permissions they must manage.

Organizations use multiple identity systems to manage internal employee identities, while other tools address customers through a customer identity and access management (CIAM) solution. Most enterprises now run more than one identity provider (IdP), and man-y run several at once. Multiple IdPs help enterprises avoid vendor lock-in, enforce tailored IAM services, and meet regulatory requirements.

But using multiple IdPs also introduces fragmentation and limited visibility issues because many tools don't connect natively. Poorly integrated tools create a fragmented environment where security teams struggle to manage the identity and access of applications. That fragmented environment breeds dormant accounts, overprivileged and residual access, and poor lifecycle management.

Machine credentials such as API keys, certificates, and AI agent identities are a separate discovery problem with its own methods. This paper focuses on privileged accounts in the username-and-password sense: the personal, shared, and service accounts every application carries.





How Application Type Determines Discovery Depth

Applications have a distinct approach to granting and storing privileges. This is why application owners should know which type they run, because the type determines how deep discovery has to go. In many environments, organizations deploy one or more of the following application types:

1. SaaS

SaaS applications are the simple case. You don't manage the infrastructure, so there are no server or database accounts to find. Discovery stops at the application itself: the admin roles and service accounts that exist at the UI and API level. The accounts are fewer, but they often carry broad access across business systems, so every one of them belongs in the inventory.

2. Cloud-hosted legacy applications

Cloud-hosted legacy applications are the moderate case: a commercial off-the-shelf product or service deployed on a cloud instance or virtual machine. The application's own admin accounts are only the start. Privileged accounts also exist at the operating system, domain, database, and enterprise-integration levels of the machine it runs on, often as service accounts or credentials sitting in configuration files. Discovery has to cover every one of those layers, up to the application UI.

3. Modern applications (containers and serverless)

Modern applications, whether containerized or serverless, are also a moderate case. There is no conventional server to scan, but privileged accounts still exist at the service layer, in the database or key manager behind the application, and up to the application UI. Discovery here means working through those layers rather than assuming cloud-native means account-free.

4. Datacenter-hosted legacy applications

Datacenter-hosted applications are the deep end. The application's own admin accounts are only the top layer; beneath them sit privileged accounts at the operating system, domain, database, and enterprise-integration levels, many of them shared administrator or Active Directory service accounts that have accumulated permissions for years. Discovery here means working through every layer the application touches, not just the login screen.

5. Homegrown legacy applications

Homegrown applications are complex for a different reason: they were built in-house, so authentication and authorization are custom, credentials often live in code, scripts, configuration files, or local databases, and documentation may not exist. Discovery covers the same layers as any hosted application, plus the work of finding what the original developers embedded along the way.

Where Do Privileged Accounts Hide in Organizations?

Privileged accounts hide in the parts of your environment routine reviews skip, from the admin accounts on the systems that host your applications to the service accounts, shared logins, and embedded credentials inside them. They fall into four groups: personal, shared, technical accounts, and orphaned accounts.



1. Privileged Personal Accounts

These are additional accounts individuals hold apart from enterprise user accounts: the domain admin account assigned to a system administrator, the local admin account an application owner uses on a server, and the elevated login created for troubleshooting and never removed. Domain admin accounts carry unrestricted control over Active Directory. Local admin accounts let users override endpoint protections, and because they are often created ad hoc, they rarely appear in any inventory.



2. Privileged Shared Accounts

These accounts don't belong to a specific individual: built-in super admins like root and administrator, break-glass and emergency accounts, and common logins used by more than one person. Because no single person owns them, their passwords are rarely rotated and their activity can't be tied to anyone. They tend to live quietly on the operating systems and databases beneath applications, exactly where directory scans don't look.



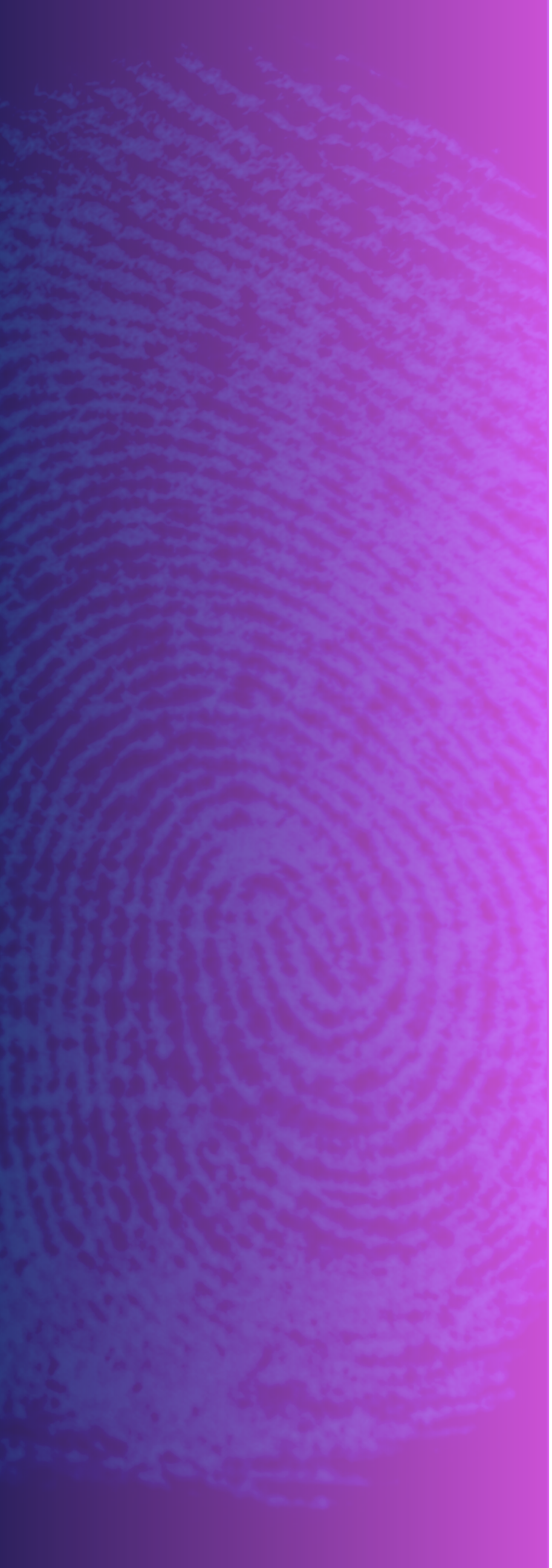
3. Privileged Technical and Service Accounts

These accounts are used by systems and processes. This category includes service accounts that run background services, scheduled tasks, and application-to-database connections, and the static credentials hardcoded into scripts and configuration files. They often require elevated privileges to function, their passwords frequently never expire, and because they operate silently, they accumulate rights for years without anyone noticing.



4. Orphaned Accounts

Accounts in any of the three classes above become orphaned when employees leave or systems are retired. They keep their privileges but lose their owner. This means no one reviews, rotates, or notices them, making orphaned accounts the most likely to stay hidden the longest. Where each of these classes lives depends on the application type. A SaaS application only carries them at the UI and API level; a hosted application carries them at every layer down to the operating system. That is why discovery has to be matched to application type rather than run as one generic scan.



How Should You Discover Privileged Accounts?

Every control that protects a privileged account starts with finding it. Discovery is the strategy, and two practices make it work: running it continuously and pairing every discovered account with an owner.

You can't protect accounts you haven't found. Application discovery helps you surface privileged accounts and credentials across directories, cloud platforms, servers, databases, SaaS admin consoles, and other areas where they exist in your environment.

Discovery should be continuous, not a one-time exercise, because new applications, services, and integrations are constantly introduced into the environment. Continuous discovery can be achieved through automated scanning tools, Privileged Access Management (PAM) solutions, identity governance platforms, and regular audits that detect new privileged accounts or changes in permissions. Ongoing discovery helps application teams maintain visibility, quickly identify risk, and keep privileged access controls aligned with the evolving environment.

Ownership should accompany discovery because it ensures accountability for managing privileged access, credentials, and accounts. Application owners validate access requirements, review privileges, and ensure regulatory compliance where necessary. Without assigned ownership, privileged accounts can become orphaned, unmonitored, and difficult to review, creating significant security risks.

Protect Privileged Accounts at Scale with Identity Orchestration

Many organizations struggle with protecting privileged accounts because the environment (identities, permissions, and application volume) is constantly changing, leaving privileged identities vulnerable to attacks. One way enterprises can protect privileged accounts is to implement an identity fabric to coordinate identities, permissions, governance, and security controls together under a common framework. The unification lets security teams discover and onboard privileged accounts across apps and cloud providers and consistently apply identity security principles and controls.

Enterprises need identity orchestration to operationalize and execute workflows in an identity fabric. The identity orchestration service coordinates disparate IAM systems to create cohesive, frictionless identity workflows across multiple systems in your environment. While identity orchestration centralizes IAM in one view, it still needs qualified hands to make it work. This is what Simeio Identity Orchestrator (IO) provides.

Simeio's role starts earlier than the platform. Engagements include the discovery itself: what to look for, where each application type hides it, and how to run the process. The scanner is the only piece organizations bring.

Simeio IO bundles orchestration services as a managed service with capabilities that expand as the need arises, while labor and capital costs stay flat. It treats discovery and onboarding as connected stages of one process rather than separate projects. Organizations bring results from the scanning and discovery tools they already use into Simeio IO, which provides a business-facing layer to score application risk and prioritize what to onboard first. From there, Simeio IO automates onboarding across multiple PAM technologies. Large organizations often run more than one PAM system, one for human access and another for secrets, and Simeio IO onboards them simultaneously without the organization needing in-house skill sets for each. Applications and their privileged accounts move from discovered to governed without manual handoffs between teams. Onboarding is what brings discovered accounts under vaulting, rotation, and least-privilege policy.

Simeio IO's self-service, no-code model also lets application owners onboard apps and configure resources with pre-designed templates and AI-assisted workflows without specialized coding skills. Simeio IO is also vendor-neutral and can help security teams translate configurations from legacy systems to modern ones, especially during migrations. This service keeps identity controls and operational workflows consistent across legacy and modern resources.

The full benefits of Simeio IO include:

- Unified control plane across Access Management (AM), IGA, CIAM, and PAM
- Business-facing application risk scoring that turns external scan results into onboarding priorities
- Automated onboarding across multiple PAM technologies
- Vendor-neutral integration with modern and legacy systems
- AI-assisted security templates for seamless configuration and identity execution
- Clear ownership assignment for onboarded applications



Move From Exposure to Durable Control

Privileged accounts are multiplying faster than most organizations can discover and onboard them, and every account left unfound is one no control can touch. Security leaders must shift from fragmented, manual oversight to unified orchestration.

Simeio IO offers a practical path to turn discovery results into governed, onboarded applications and consistently apply identity policies at scale. Security teams that wait until privilege creep becomes a breach have waited too long. Orchestration is how they get control back.

Onboard your applications and their privileged accounts for protection with Simeio IO >

KEY TAKEAWAYS

The real gap is between discovery and onboarding: Every new application brings privileged accounts with broad standing access, and many are never onboarded. Accounts stuck in that gap stay unvaulted, unrotated, and unmonitored, which fuels privilege creep.

Fragmentation widens privilege creep: Multiple IdPs and poorly integrated tools limit visibility, creating dormant, overprivileged, and orphaned accounts that routine reviews miss.

Application type determines protection depth: SaaS, cloud-hosted legacy, modern containerized and serverless, datacenter-hosted legacy, and homegrown applications each store and grant privilege differently. Matching discovery methods and controls to each type closes blind spots that a one-size-fits-all approach leaves open.

Discovery plus ownership is the core control: Continuous discovery surfaces privileged accounts as applications change, and an assigned owner keeps each account reviewed, accountable, and ready to onboard.

Orchestration makes protection sustainable at scale: An identity fabric unifies disparate IAM systems, and identity orchestration operationalizes consistent policies across legacy and modern applications. Simeio IO delivers orchestration as a managed, vendor-neutral service, helping organizations move from fragmented oversight to durable control without incurring labor and capital costs.

FAQs

How do we find privileged and standing-access accounts we don't even know exist?

Start with the application, not the directory. A directory scan surfaces the accounts you already half-know about; the ones you miss live at the other layers of each application: the operating system, database, and integrations behind it. Classify each application by type, search the layers that type carries, and assign an owner to what you find so the inventory stays current instead of going stale after one audit.

Does Simeio Identity Orchestrator (IO) replace our existing scanning and discovery tools?

No. Simeio IO does not perform the scanning itself. Organizations bring results from the scanning and discovery tools they already use into Simeio IO, which treats discovery and onboarding as connected stages of one process rather than separate projects.

How does knowing my application type change how I protect privileged accounts?

Each application type carries privileged accounts at different layers. A SaaS application only exposes accounts at the application UI and API level, since you don't manage its infrastructure. A hosted application, whether in the cloud or your data center, carries privileged accounts at the operating system, domain, database, and enterprise-integration levels as well as in the application itself. Homegrown applications add custom authentication and credentials stored in code or configuration. Knowing the type tells you how deep discovery has to go.

Why can't our existing IAM tools handle privileged account discovery, and what does identity orchestration add?

The problem is that your IAM tools don't talk to each other. Fragmentation across AM, IGA, PAM, and cloud resources lets privileged accounts accumulate unseen. Identity orchestration adds a coordination layer that operationalizes an identity fabric, applying discovery and onboarding consistently across those disconnected systems from one control plane. Simeio IO delivers this as a managed service, unifying legacy and modern tools without ripping out existing investments, so you close the gaps without adding headcount.

What role do application owners play in protecting privileged access?

Application owners provide the accountability that discovery alone cannot. They validate which access is actually required, review and recertify privileges, and help ensure regulatory compliance. Without an assigned owner, privileged accounts become orphaned and unmonitored, making them difficult to review and attractive targets for attackers.