Teleport

Automation is the New Attack Surface

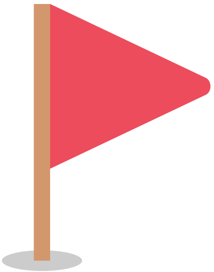
Securing Non-Human Identities (NHIs) at the Infrastructure Layer

Contents

The Unseen Layer of Risk

Modern infrastructure moves fast. Automation now powers nearly every critical system — from provisioning environments with code, to deploying software through pipelines, to scaling workloads across distributed services and AI agents.

But beneath that velocity lies an expanding layer of risk that's often invisible: non-human identities (NHI).



These identities — **bots, scripts, CI jobs, deployment agents, service accounts** — are doing the real work of delivering infrastructure. Yet most operate outside the boundaries of modern identity governance. They authenticate with static tokens or embedded secrets, often carry excessive permissions, and rarely expire. There's no session tracking, no behavioral visibility, and little alignment with Zero Trust frameworks.

While most organizations have strengthened security for human access with SSO, MFA, and approval-based workflows, automation remains unauthenticated. And the consequences are unfolding across enterprise environments and major service providers.

A compromised CI job can leak thousands of secrets across environments. A single token stolen from a developer's machine can lead to multi-cloud data exfiltration. An AI agent operating on a privileged key can leak massive amounts of internal data without oversight. These are recurring patterns in recent breaches.

The cost of inaction is compromised security, operational complexity, audit failure, regulatory exposure, and a growing disconnect between what security teams believe is governed and what automation can actually reach.

Fragmented by Design: The Non-Human Identity (NHI) Problem

Most security teams assume their non-human identities are secured. There's a secrets vault in place, IAM roles configured, maybe even some policy-as-code.

But under the surface, non-human identities operate via a patchwork of disconnected tools and security configurations that don't hold up in distributed environments at scale.

Machine and workload credentials are scattered across secrets managers, CI environments, YAML files, and cloud IAM systems.

None of these tools provide unified visibility into how identity is issued, who—or what—is using them, or when they should be revoked

Secrets are often provisioned manually, shared across services, and rarely tied to a lifecycle.

IAM roles are static, over-permissioned, and misaligned across clouds.

Federation efforts tend to be brittle and require constant upkeep.

Policy frameworks often validate infrastructure configuration, but not the automation that's applying it.

These gaps lead to systemic failure: credential sprawl that attackers can exploit, orphaned access that no one remembers to clean up, and audit processes that depend on guesswork and log correlation. As environments scale, the operational burden of managing secrets, access roles, and trust relationships balloons, slowing down teams while widening the blast radius.

At the same time, business demands are increasing: delivery must accelerate, security must improve, and regulatory standards must be met. Zero Trust principles are being mandated at the human layer; however, automation still runs on static trust assumptions, long-lived credentials, and hardcoded secrets.

Solving this requires a shift from fragmented tooling to a unified model that governs identities at runtime: how they're issued, enforced, and observed. That model needs to work across environments, map to existing workflows, and establish machines and workloads as cryptographically verifiable identities.

What Breaches Are Telling Us

Over the past two years, some of the most damaging incidents in cloud infrastructure didn't start with a compromised user. They began with automation: a leaked token, a trusted service account, or a compromised CI workflow.

!

An AI research team at a major cloud provider exposed 38 terabytes of internal data — including credentials and internal communications — through an ungoverned SAS token embedded in an AI dataset URL shared in GitHub.

A small mistake, made in an automated system, led to massive lateral exposure.

!

Elsewhere, Terraform automation tokens were pushed to public repositories with full organizational access, while a second-party service integration exposed service account credentials, triggering unauthorized access across customer environments.

These failures didn't come from malicious insiders; they came from trusted systems doing exactly what they were configured to do, without sufficient guardrails.

Without session identity or traceability, teams have no idea what those credentials were used for. Without centralized access control, they can't contain the blast radius quickly. And because these systems operate silently and continuously, the window for detection is slim — if it exists at all.

These are systemic gaps in how organizations manage access for automated systems that exist in almost every environment:

- Credentials reused across systems
- Automation with persistent, overpermissioned access
- No ability to audit or revoke machine behavior

The impact of these incidents extends into audit and compliance risks, developer workflow delays, and multi-week investigation cycles that pull teams away from shipping value.

The automation layer has become the breach entry point. And until NHI are governed with rigor, that door stays open.

From Static Secrets to Infrastructure Identity

Infrastructure Identity defines how automated systems authenticate, gain access, and leave traceable footprints, without relying on static credentials, hardcoded tokens, or cloud-specific workarounds.

- Instead of embedding secrets or assigning long-lived roles, systems are issued cryptographic identities with short-lived, policy-bound privileges at runtime.
- Access is least privileged, time-limited, and tied to a unique workload identity.
- When the task ends, the identity expires — no secrets rotation, no sprawl.

This model eliminates orphaned access and reduces the attack surface. It also makes revocation immediate and auditability effortless. Every action is traceable to a specific identity, with no ambiguity about who or what touched critical systems.

Infrastructure Identity simplifies security across the CI/CD stack:

- Pipelines can deploy without storing secrets.
- Infrastructure-as-code jobs apply changes using time-bound permissions.
- AI agents operate with least privileged access and leave behind a complete audit trail.
- Cross-cloud authentication becomes consistent and centrally governed.

Organizations adopting Infrastructure Identity with Teleport are securing multi-cloud service orchestration, enforcing policy boundaries for AI systems running in production, and meeting regulatory requirements for identity-bound auditability across frameworks like PCI, FedRAMP, and ISO 27001. Teleport strengthens both security posture and operational clarity, giving teams confidence in every automated action their systems take.

Teleport applies the model of Infrastructure Identity across every stage of automation, integrating access and audit into the infrastructure itself.

Deploying Infrastructure Identity with Teleport

Teleport delivers unified cryptographic identity for both people and non-human identities — hardening security, unifying access control with governance, and eliminating the operational overhead of legacy credential management.

At the core of Teleport Machine & Workload Identity is `tbot`, a lightweight agent that issues short lived identity to machines and workloads at runtime using cryptographic certificates. Each identity is tied to a role and associated privileges, and expires automatically. Nothing is hardcoded. Nothing lives forever. And every system, service, or job can be traced back to a cryptographic identity.

Access is policy-driven and time-bound. Whether it's a CI/CD workflow deploying infrastructure, an internal agent querying a database, or a cloud-native service invoking an API, each machine or workload identity is granted only the access it needs, for only as long as it needs it. When access is no longer required, it's gone automatically.

Teleport logs every action tied to a machine or workload identity: every command, database query, session, and request. Those logs are centralized, searchable, and bound to the identity that triggered them, giving teams full visibility and fast answers during audits or incident response.

This identity-first model works across cloud providers, Kubernetes clusters, legacy infrastructure, and SaaS endpoints, enabling consistent governance and real-time observability, even in complex, hybrid environments. The result is machine and workload identity that's secure by default, fully auditable, and aligned with how infrastructure actually operates.

With Teleport, identities are tied to a role, scoped by policy, and expire automatically.

Nothing is hardcoded. Nothing lives forever.

And every system, service, or job can be traced back to a cryptographic identity.

The Future State of Machine and Workload Identity

Infrastructure teams are shifting from managing secrets to governing identity. Instead of provisioning cloud keys, rotating tokens, or locking down service accounts manually, they issue short-lived identity at runtime, with least privileged access determined by role and policy, and which automatically expires when the job is done.

CI/CD jobs no longer carry long-lived credentials. When a workflow runs, it requests a short-lived identity via `tbot` that grants access to exactly what's needed, like deploying to a specific environment or querying a particular API. The certificate then expires with nothing stored in environment variables or version control.

Cloud accounts that once held static credentials are now replaced with ephemeral identities that authenticate through short-lived, contextual access. These identities are automatically mapped to roles, governed through a policy engine that manages permissions across Kubernetes clusters, cloud services, databases, and internal systems.

Audit logs provide full attribution. Security and platform teams can see exactly which identity accessed what infrastructure, at what time, and under what context, without stitching together logs from CI, IAM, and secrets managers.

During incident response, that visibility becomes leverage. Teams can pinpoint the behavior of a specific workload, job, or agent and revoke its access instantly, without coordinating across systems, triggering mass credential rotations, or shutting down operations.

And most importantly, automation moves faster. Infrastructure changes are deployed without credential provisioning delays. Secret rotation no longer causes breakage. Access is granted automatically and precisely, based on cryptographic identities with contextual privileges.

This is how organizations scale securely: by giving machines first-class identity, and by treating access as something that's provisioned, governed, and expired by design.

Turning Machine Access Into a Strategic Advantage

Automated systems now deploy code, provision infrastructure, and move data between environments without human intervention. Yet most organizations still lack clear governance over how these systems access critical resources.

By securing access for automated systems through identity, teams reduce breach risk, simplify audits, and eliminate the need to manage secrets manually. Access adheres to the principle of least privilege and is granted only for the duration of a task. Every interaction is traceable. And platform teams can operate with speed, knowing that controls are in place by default.

Teleport provides the foundation for this shift: short-lived credentials, unified policy enforcement, and visibility across every cloud, environment, and protocol.

Move faster. Reduce risk. Prove control.

- Issue cryptographic identity at runtime, without storing secrets
- Enforce least privileged, just-in-time access across machines, environments, and teams
- Trace every automated action to a specific identity, policy, and purpose