

HIGHPOINT

2026 The Non-Human Identity Frontier Whitepaper

info@highpoint.com | HighPoint.com

Table of Contents

3	Executive Summary
3	Problem Statement
4	The Statistical Divergence of Human and Machine
4	The Inherent Vulnerability of Non-Human Credentials
5	The AI-Driven Risk Multiplier
5	Technical Background
6	The Anatomy of an NHI: Consumer, Secret, and Entitlement
6	The Evolution of Identity Lifecycle Management
7	Distinguishing M2M Authentication Methods
7	The AI Risk Multiplier: A Deep Dive into Agentic Threats
8	The Credential Expansion Problem
8	Proposed Solution: The Non-Human Identity Management (NHIM) Framework
8	Novel Attack Vectors in AI-Agent Environments
9	The Pillars of NHIM
10	Implementation Details: A Technical Roadmap
11	Benefits & Use Cases
12	Challenges & Mitigation Strategies
13	Analysis of Real-World Breaches: Lessons Learned
13	Core Insights from Breach Data
14	Conclusion

Executive Summary

The digital landscape of 2025 is characterized by a silent but profound demographic shift. While cybersecurity strategies have historically centered on the human user—the “weakest link” in the security chain—the primary actors within the modern enterprise are no longer human. The rapid transition to cloud-native architectures, microservices, and autonomous agentic artificial intelligence has precipitated an explosion of non-human identities (NHIs). These entities, which include service accounts, API keys, OAuth tokens, secrets, and certificates, have quietly become the invisible backbone of global infrastructure.

Recent research reveals a staggering trend: non-human identities now outnumber human users by a ratio of approximately 144 to 1, representing a 44% increase over the previous year. This proliferation is not merely a statistical curiosity but a systemic vulnerability. Unlike their human counterparts, NHIs frequently operate with elevated privileges, lack multi-factor authentication (MFA), and exist outside the structured lifecycle governance provided by human resources and identity management systems. Furthermore, the emergence of agentic AI acts as a significant risk multiplier. A single autonomous AI agent can require 15 to 20 distinct non-human identities to function across integrated environments, creating a new identity frontier that traditional security models are ill-equipped to manage.

This white paper provides a comprehensive analysis of the NHI ecosystem. It details the technical anatomy of these identities, the catastrophic risks posed by their

unmanaged sprawl, and the systemic failures of current identity and access management (IAM) strategies. We propose a robust Non-Human Identity Management (NHIM) framework rooted in Zero Trust principles, focusing on automated discovery, secretless authentication, and continuous lifecycle governance. By treating NHIs as first-class citizens in the security hierarchy, organizations can mitigate the exponential threats posed by automation and safely harness the power of the AI-driven economy.

Problem Statement

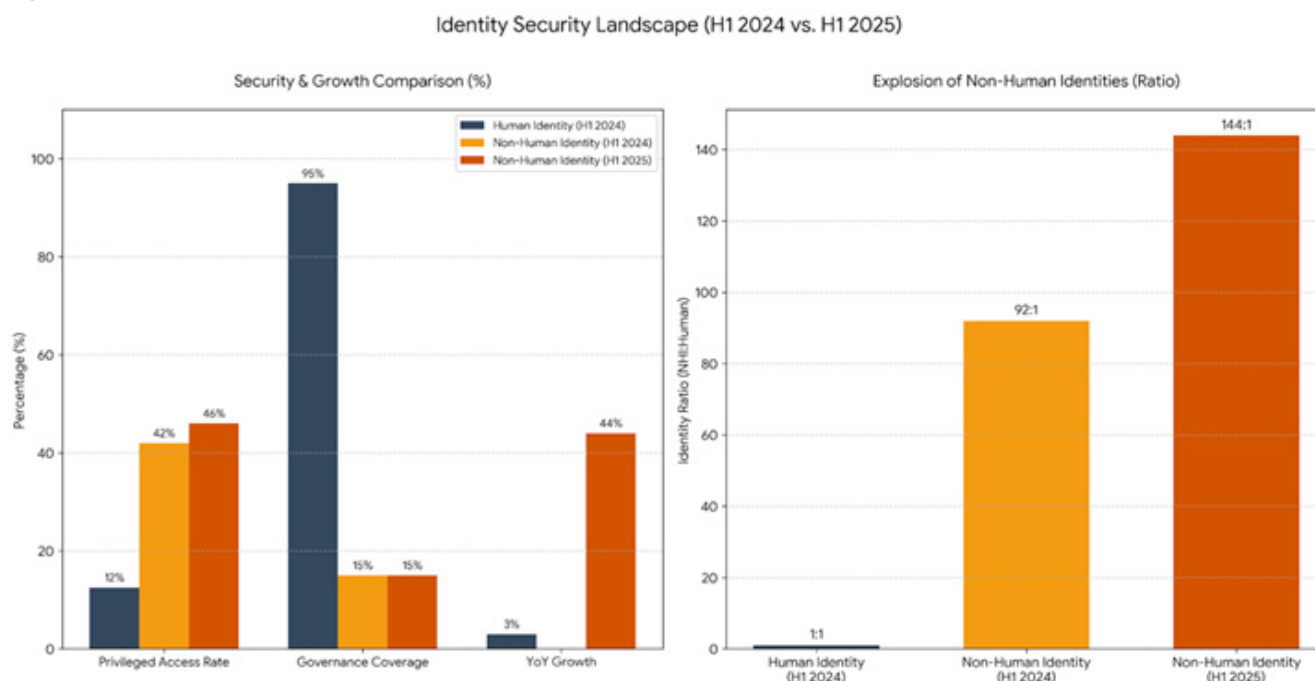
The central challenge facing modern IT and security teams is the “under-the-radar crisis” of non-human identity sprawl. For decades, the industry perfected the management of human digital identities through Joiner-Mover-Leaver (JML) processes, single sign-on (SSO), and biometric authentication. However, the move toward DevOps, cloud-native deployments, and now agentic AI has introduced a new class of identity that lacks these fundamental safeguards.

The Statistical Divergence of Human and Machine

The scale of the problem is best understood through the lens of recent enterprise data. While the number of human employees in an organization grows linearly with business expansion, the number of non-human identities grows exponentially with the deployment of every new microservice, container, and automated workflow.

This growth creates a “visibility gap” where security teams are unaware of up to 94% of the service accounts and API keys active within their environment. This lack of visibility is particularly dangerous because NHIs are frequently over-privileged. Research indicates that 97% of non-human identities have excessive permissions, often granted by developers for the sake of convenience or to avoid breaking critical automated pipelines. (Figure 1)

Figure 1



The Inherent Vulnerability of Non-Human Credentials

The technical nature of non-human identities makes them inherently more difficult to secure than human identities. Humans can use MFA, respond to behavioral prompts, and are subject to physical verification. NHIs, by contrast, rely on static secrets—passwords, keys, or tokens—that are “consumed” programmatically. If an attacker steals a human’s password, they still face the hurdle of MFA. If an attacker steals

a service account’s API key, they gain unfettered, legitimate-appearing access to the system.

Common failures in the management of these credentials include:

- **Hardcoding in Source Code:** API keys and tokens are frequently committed to GitHub or GitLab repositories where they can be scraped by automated bots.
- **Plaintext Storage:** Secrets are often stored in unsecured environment files or configuration files on servers.

- **Lack of Rotation:** 71% of non-human identities are not rotated within recommended time frames, allowing compromised credentials to remain useful to attackers for months or even years.
- **Orphaned Identities:** When a project is decommissioned, its associated service accounts often remain active, creating “zombie” entry points for lateral movement.

The AI-Driven Risk Multiplier

The recent surge in agentic AI deployment represents a critical tipping point. Unlike traditional software that follows rigid logic, autonomous AI agents make independent decisions and interact with a vast array of tools and databases. Each interaction requires an identity. As organizations shift from singular LLMs to multi-agent systems, the volume of required credentials shifts from linear to exponential. These agents often require high-velocity authentication, frequently spanning multiple cloud providers and third-party APIs, creating an environment where traditional audit trails and manual oversight collapse.

Technical Background

To develop a robust solution, one must first understand the fundamental technical composition of a non-human identity and how it differs from traditional human-centric constructs. At its core, an NHI is a digital entity assigned to a machine, application, or process to facilitate automated communication and resource access.



The Anatomy of an NHI: Consumer, Secret, and Entitlement

Every non-human identity functions through the interaction of three primary technical components, often referred to as the “NHI Triad”.

- 1. The Consumer: This is the “who” of the NHI. It represents the application, script, bot, or virtual machine that initiates an action. In cloud-native environments, the consumer could be a Lambda function, a Kubernetes pod, or an AI agent.
- 2.The Secret: This is the “proof” of the identity. It is the programmatic credential used to authenticate the consumer to a target resource. Common secrets include API keys, JSON Web Tokens (JWT), OAuth 2.0 bearer tokens, and X.509 digital certificates.
- 3.The Entitlement: This represents the “what.” It is the set of permissions or roles granted to the identity. In an AWS environment, this might be an IAM policy; in a Kubernetes cluster, it is defined by Role-Based Access Control (RBAC)

The Evolution of Identity Lifecycle Management

Historically, identity management was a human-centric discipline. The “Joiner-Mover-Leaver” (JML) process was driven by human resource (HR) events. When an employee joined, they were provisioned an account; when they left, the account was revoked.

For non-human identities, the governing process is not HR but Change Management. NHIs are created during development, deployment, or configuration updates. Because they do not “resign” or “retire” in a human sense, they often lack a defined end-of-life trigger. Without integration into a Configuration Management Database (CMDB) or a dedicated NHIM system, these identities persist indefinitely, contributing to the digital sediment that attackers exploit. (figure 2)

Figure 2

Component	Technical Example	Security Vulnerability
Consumer	AWS EC2 Instance	Can be compromised via OS vulnerabilities
Secret	OpenAI API Key	Can be leaked in code or logs
Entitlement	s3:FullAccess	Often violates the Principle of Least Privilege

Distinguishing M2M Authentication Methods

Technical stakeholders must navigate a complex landscape of authentication methods, each with varying security postures. (Figure 3)

Figure 3

Method	Mechanism	Risk Level
Static API Keys	Shared secret string	High: Long-lived, easily leaked.
Service Accounts	Workload Identity	High: Long-lived, easily leaked
OAuth 2.0 Tokens	Delegated authority tokens	Medium: Short-lived but can be intercepted.
MTLS Certificates	Mutual cryptographic proof	Low: Strong, but difficult to rotate at scale.
OIDC / Federation	Trust-based token exchange	Lowest: Enables secretless authentication.

The AI Risk Multiplier: A Deep Dive into Agentic Threats

The introduction of agentic AI represents more than just a volume increase; it is a qualitative shift in the threat landscape. Traditional automation is deterministic—if X occurs, perform Y. Agentic AI is non-deterministic, utilizing reasoning loops (e.g., LLM -> Plan -> Tool Use -> Reflect) to achieve goals. This can lead to AI driven processes being created at exponential rates to meet the required data driven objective.



The Credential Expansion Problem

As organizations deploy autonomous agents, the number of required NHIs expands exponentially. For example, a customer service AI agent might require:

1. A primary authentication credential to identify the agent itself.
2. Unique API tokens for every integrated SaaS tool (e.g., Salesforce, Zendesk, Stripe).
3. Database connection strings to access customer history.
4. Notification service credentials (e.g., Twilio, SendGrid) to communicate with users
5. Observability credentials to log activity for debugging.

A single agent deployment can thus create an ecosystem of 15–20 distinct NHIs. If an enterprise deploys 100 specialized agents, they have suddenly introduced 2,000 new credentials into their environment, often without the same scrutiny applied to a single human hire.

Novel Attack Vectors in AI-Agent Environments

Agentic AI introduces vulnerabilities that traditional security firewalls are unable to perceive:

- Indirect Prompt Injection: An attacker does not need to compromise the agent directly. Instead, they can place malicious instructions in a document or database the agent is programmed to read. This “injection” can trick the agent into using its legitimate NHI credentials to exfiltrate data or modify systems.
- Recursive Delegation Risks: Agents can spawn sub-agents to complete complex tasks. If the parent agent passes its high-privilege credentials to an unvetted sub-agent, the organization loses the chain of custody and accountability.
- Credential Caching in LLM Memory: Many AI SDKs (Software Development Kits) from providers like OpenAI or Anthropic require API keys at initialization. These secrets are often stored in the memory space of the running model, where they can be exfiltrated if the model itself is hijacked.

Proposed Solution: The Non-Human Identity Management (NHIM) Framework

To address the exponential risks of NHI sprawl and AI-driven threats, organizations must move beyond fragmented point solutions and adopt a unified Non-Human Identity Management (NHIM) framework. This framework is designed to govern the entire lifecycle of an NHI, from discovery and provisioning to continuous monitoring and decommissioning.

The Pillars of NHIM

The proposed solution is built upon four fundamental technical pillars:

1. Continuous Discovery and Visibility

The foundation of NHIM is “Holistic Contextual Visibility”. Organizations cannot secure what they cannot see. A robust NHIM solution must automatically discover all workloads, service accounts, and secrets across the hybrid cloud, including those “shadow” identities created by developers outside of standard workflows. This discovery must extend into code repositories (GitHub, GitLab), CI/CD pipelines (Jenkins, GitHub Actions), and collaboration tools (Slack, Jira), where secrets often leak.

2. Automated Lifecycle Governance

NHIs must be managed with the same rigor as human employees. The NHIM framework automates the provisioning, rotation, and retirement of credentials.

- Ownership Attribution: Every NHI must be mapped to a human owner or team responsible for its lifecycle.
- Just-in-Time (JIT) Provisioning: Rather than having static, “standing” privileges, NHIs should be granted permissions only when needed for a specific task, which are then revoked upon completion.
- Automated Decommissioning: When a service or project is terminated, the NHIM system must automatically identify and disable the associated NHIs to prevent the creation of orphaned accounts.

3. Transition to Secretless Authentication

The “gold standard” of NHIM is the elimination of long-lived, static secrets entirely. This is achieved through workload identity federation using standards like SPIFFE (Secure Production Identity Framework For Everyone) or OpenID Connect (OIDC).

- Short-Lived Ephemeral Tokens: Workloads exchange cryptographic proofs for short-lived access tokens, meaning there are no permanent API keys to steal.
- Managed Identities: In cloud-native environments, organizations should prioritize Azure Managed Identities or AWS IAM Roles for Service Accounts, where the cloud provider handles the underlying credential rotation invisibly.

4. Behavioral Monitoring and Threat Detection (NHIDR)

Because NHIs behave predictably, their actions can be baselined. Non-Human Identity Detection and Response (NHIDR) systems monitor for deviations from these baselines. For example, if a service account that normally reads 10 database records a minute suddenly attempts to export 100,000 records, the system can automatically quarantine the identity and alert the security operations center (SOC).

Implementation Details: A Technical Roadmap

Implementing a NHIM framework is a multi-stage process that requires coordination across development, security, and operations teams.

Stage 1: Inventory and Risk Assessment

The first step is to establish a “Single Source of Truth” for all non-human identities.

Stage 2: Hardening and Governance Enforcement

Once visibility is achieved, the focus shifts to reducing the attack surface.

- Eliminate Exposed Secrets: Remove hardcoded keys from code and configuration files, moving them into secure vaults.

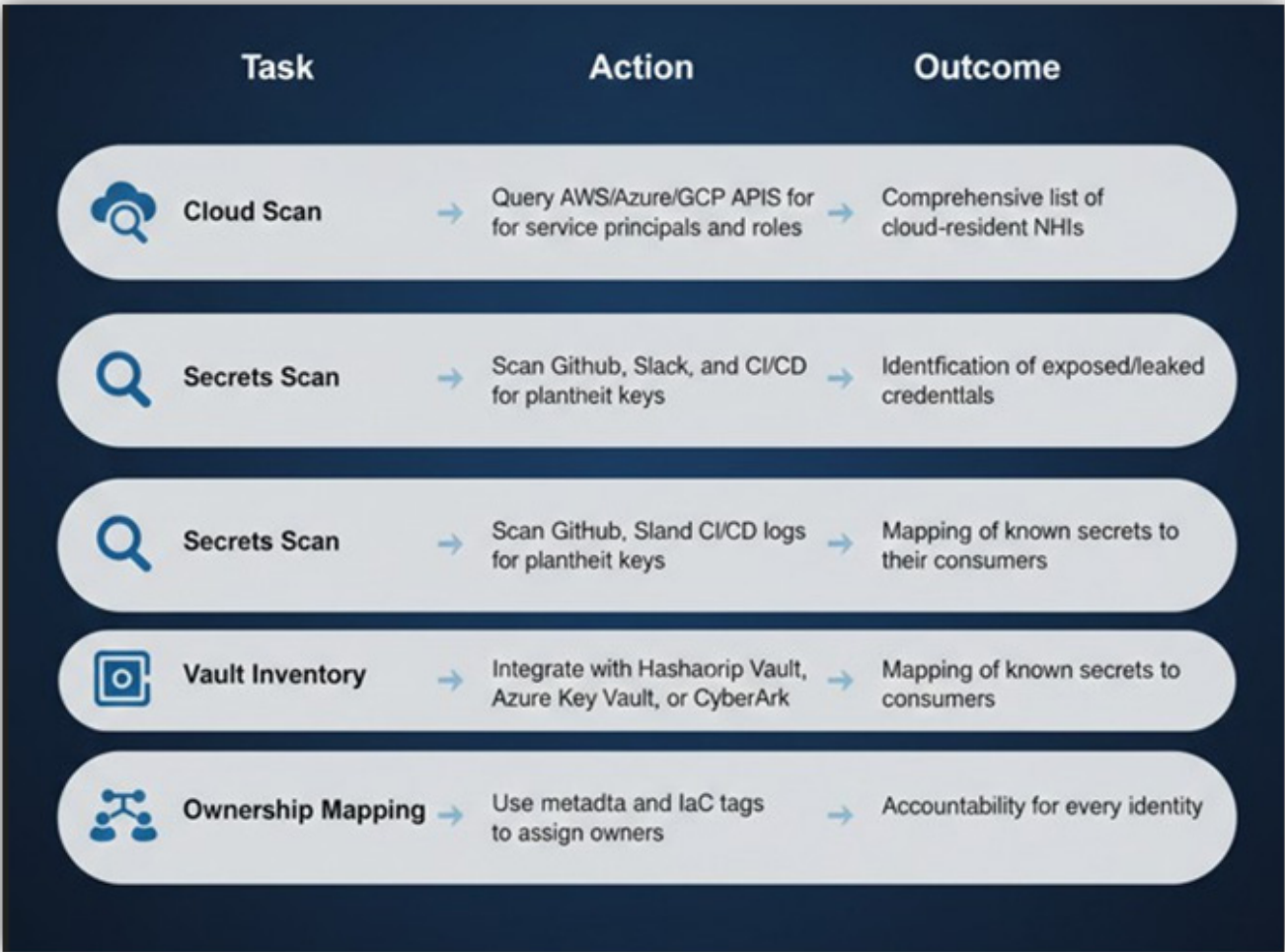
- Least Privilege Remediation: Analyze the permissions used by an NHI over a 30-90 day period and shrink its scope to only those required.
- Establish Naming Conventions: Implement structured naming for service accounts (e.g., svc-prod-api-gateway-east) to enable automated policy enforcement and simplify incident investigations. (Figure 4)

Stage 3: Operationalizing Lifecycle Automation

The final stage involves moving toward a fully automated secretless target state.

- Integrate with CI/CD: Build NHI creation and rotation into the deployment pipeline. When a new microservice is deployed, its identity should be provisioned automatically with a short lifespan.

Figure 4



- **Automated Rotation Schedules:** For remaining static secrets, implement automated rotation. If an API key is not rotated within 30 days, the NHIM system should trigger a workflow to update it.
- **AI Agent Guardrails:** For agentic AI, implement an Identity Control Plane that fingerprints each agent and enforces runtime policies, such as limiting the types of APIs an agent can call or the sensitivity of data it can access.

Benefits & Use Cases

A comprehensive NHIM framework provides tangible benefits beyond simple risk reduction, enhancing operational efficiency and enabling innovation in complex environments.

Use Case 1: Securing the DevOps Pipeline

In a modern CI/CD (Continuous Integration/Continuous Deployment) environment, automated tools must move code from development to production. This requires broad access to repositories, build servers, and cloud infrastructure.

- **The Challenge:** Traditional static tokens in Jenkins or GitHub Actions are a primary target for attackers. In 2024, attackers compromised a GitHub Action to exfiltrate secrets from over 23,000 repositories.
- **The NHIM Benefit:** By using ephemeral, JIT credentials for build jobs, the credentials exist only for the duration of the deployment. If an attacker gains access to the build server after the job is done, they find no useful credentials to steal.

Use Case 2: Financial Services and Algorithmic Trading

Banks and trading firms rely on automated algorithms that perform thousands of trades per second, often using machine learning models.

- **The Challenge:** These systems require access to sensitive financial data and trading platforms. A compromised NHI could be used to execute fraudulent trades, leading to systemic financial loss.
- **The NHIM Benefit:** NHIM provides the granular control and real-time monitoring needed to secure these high-stakes identities. It ensures that only authorized algorithms can access specific market data feeds and provides a tamper-proof audit trail for regulatory compliance (e.g., SOC2, PCI DSS).

Use Case 3: Industrial IoT and Healthcare

In the “Internet of Medical Things” (IoMT), thousands of devices—from infusion pumps to EKG monitors—must communicate with hospital portals and physicians.

- **The Challenge:** Many of these devices have limited computing power and use hardcoded credentials that are never changed. A compromised pacemaker or insulin pump could have life-threatening consequences.
- **The NHIM Benefit:** NHIM facilitates the enrollment and provisioning of unique, certificate-based identities at IoMT scale. It ensures that only verified devices can upload patient data and allows for the automated revocation of any device that shows signs of tampering.

Use Case 4: Enabling Secure AI-Driven Business Processes

As enterprises deploy “AI co-workers,” these agents will handle business processes from procurement to customer support.

- The Challenge: AI agents need to talk to dozens of SaaS tools. If each tool uses a static API key, the risk of “secret sprawl” becomes unmanageable.
- The NHIM Benefit: An identity-first architecture allows agents to use federated identities. Instead of giving an agent a permanent Salesforce key, the NHIM system issues a one-time token for a specific customer query. This limits the “blast radius” of any single interaction.

Challenges & Mitigation Strategies

Implementing NHIM is a significant strategic investment that requires addressing both technical and organizational hurdles.

Challenge 1: The Scale and Visibility Gap

With NHIs outnumbering humans 144 to 1, most organizations suffer from “visibility debt”.

- Mitigation Strategy: Do not attempt a manual inventory. Utilize automated discovery platforms that leverage cloud-native APIs and continuous scanning of code, logs, and communication channels. Prioritize high-risk environments (e.g., production VPCs) first.

Challenge 2: “Fear of Breaking” and Operational Friction

Security teams are often hesitant to rotate service account credentials or shrink permissions for fear of causing a production outage.

- Mitigation Strategy: Implement “Simulated Remediation.” Modern NHIM platforms can simulate a reduction in privileges to see if any legitimate calls would be blocked before the change is actually enforced. Additionally, utilize “Canary Rotation” for high-availability systems.

Challenge 3: Lack of Standardization and Fragmentation

NHIs exist in silos—AWS has its own system, Azure another, and SaaS providers yet another. This makes unified governance difficult.

- Mitigation Strategy: Adopt an “Identity Fabric” approach that abstracts identity management away from specific cloud providers. Use a centralized NHIM platform that integrates with all major CSPs, secret managers (HashiCorp, CyberArk), and SIEMs to provide a single pane of glass for all identities.

Challenge 4: Human Misuse of Non-Human Accounts

Developers sometimes use service account credentials to manually access systems, bypassing the MFA and logging applied to their personal accounts.

- Mitigation Strategy: Strictly enforce “Non-Interactive Access” for service accounts. If a service account is used for an interactive login, the NHIM system should automatically trigger an alert and require human validation.

Analysis of Real-World Breaches: Lessons Learned

Examining recent incidents underscores the catastrophic potential of compromised non-human identities and provides a roadmap for remediation. (Figure 5)

Figure 5

Organization	Year	Breach Mechanism	Impact
Dropbox Sign	2024	Compromised service account and API keys.	Unauthorized access to customer data and emails.
Microsoft (HaaS)	2024	Stolen Azure OpenAI API keys.	Attackers generated harmful AI content at scale.
Cloudflare	2023	Failure to rotate service account credentials.	Persistent attacker access to internal systems.
CircleCI	2022	Stolen session cookie from an engineer's laptop.	Attackers accessed customer API tokens and GitHub keys.
SolarWinds	2020	Compromised service account in CI/CD pipeline.	Global supply chain attack via poisoned software updates.

Core Insights from Breach Data

A synthesis of these incidents reveals three recurring failure patterns:

1. Persistence Through Static Credentials: In almost every case, the attackers did not “break in”; they “logged in” using a stolen service account or API key that lacked MFA and had not been rotated for months.
2. Unchecked Lateral Movement: Once inside, attackers used the over-privileged nature of service accounts to jump from development environments to production databases.
3. Supply Chain Propagation: Attacks targeting CI/CD NHIs allow a single compromise to affect thousands of downstream customers, making NHI security a public safety concern for the digital economy.

Conclusion

The evolution of the enterprise into a highly automated, AI-driven entity has rendered traditional, human-centric security models insufficient. The non-human identity is no longer a niche technical concern; it is the fundamental unit of access and risk in modern infrastructure. With NHIs now outnumbering human users 144 to 1 and growing at a rate of 44% annually, the attack surface has expanded beyond the capacity for manual human oversight. The emergence of agentic AI further accelerates this trend, introducing thousands of ephemeral, highly privileged identities that operate with a degree of autonomy previously unseen in software systems. As demonstrated by the recent surge in identity-centric breaches, the cost of inaction is severe, ranging from multi-million dollar data thefts to global supply chain compromises. To secure this new frontier, organizations must commit to a multi-year strategy focused on Non-Human Identity Management (NHIM). This requires moving toward a “secretless” architecture, enforcing the Principle of Least Privilege through automated right-sizing, and establishing continuous visibility into the entire identity landscape. By integrating NHIs into a unified security fabric and treating them as first-class citizens, enterprises can close the “visibility gap,” neutralize the exponential risks of AI-driven sprawl, and build a resilient foundation for the next decade of digital transformation. The era of the “human-only” firewall is over; the future of cybersecurity belongs to those who can master the management of the machines.

