ConductorOne

Non-Human Identity Management Defined and Explained

What are non-human identities?

Non-human identities (NHIs) are the digital credentials and permissions assigned to automated actors like machines, software, and background processes. Think of them as unique “social security numbers” for these entities, enabling them to securely prove their identity when communicating and sharing information.

NHIs are critical to the smooth operation of cloud services, automated processes, and service-oriented architectures, ensuring that only authorized entities interact with each other. They function much like human usernames and passwords but for machines and software.

What is non-human identity management?

Non-human identity management is the process of organizing and protecting these unique IDs. It involves verifying each machine identity, managing their access rights, and creating a central oversight system that ensures that every device or software has a place and role within an IT ecosystem.

Effective NHI management is crucial not only for technical reasons but also for cybersecurity. By maintaining strict control over these identities, organizations can prevent unauthorized access, data breaches, and other malicious activities.

Differences between human identities and non-human identities

Human identities are generally more centralized and stable, and ideally protected by multiple layers of security. NHIs, on the other hand, are decentralized, numerous, and often rely on single “secrets” for access, making them a major security concern.

- → Human identities are typically managed by IT or security teams, while NHIs are often created by developers, potentially leading to security gaps.
- → Human identities are easier to manage centrally, while NHIs are more decentralized, making it difficult to track and control access.
- → NHIs vastly outnumber human identities, making them a prime target for attacks.
- → Human identities are tied to individuals, while NHIs are often shared, bypassing traditional security measures.
- → Humans are authenticated using passwords, multi-factor authentication, biometrics, etc., while NHIs primarily use “secrets” like keys or access tokens, which, if compromised, can provide bad actors with unfettered access.
- → Large numbers of NHIs are constantly being created and deleted, making management challenging, while some NHIs remain unchanged for extended periods, increasing security risks.

Examples of non-human identities

Devices

- → **IoT devices:** Smart home gadgets, wearables, industrial sensors, and connected vehicles all have unique identities to authenticate and communicate with cloud platforms or other devices.
- → **Embedded systems:** Chips within appliances, medical devices, or industrial equipment often have embedded credentials for secure firmware updates or data transmission.
- → **Mobile devices:** Though they’re used by humans, mobile devices have separate identities for background tasks like push notifications, app updates, or location tracking.

Software-defined infrastructure (SDI)

- → **Virtual machines (VMs):** Each VM has its own identity for accessing network resources, storage, and interacting with other VMs in a virtualized environment.
- → **Containers:** Similar to VMs, containers have identities to manage network access, storage, and inter-container communication in a dynamic, scalable infrastructure.
- → **Network devices:** Software-defined networking relies on NHIs for routers, switches, and firewalls to authenticate and enforce security policies.

DevOps tools

- **CI/CD pipelines:** Automated build and deployment tools use NHIs to access code repositories, trigger builds, and deploy applications to various environments.
- **Configuration management tools:** Tools like Ansible or Puppet use NHIs to authenticate with managed servers and apply configurations.
- **Monitoring and logging tools:** These tools need NHIs to collect data from various systems and send alerts or reports.

Service accounts

- **Cloud service accounts:** Applications or services use cloud service accounts to interact with cloud platforms like AWS, Azure, or GCP.
- **Database service accounts:** These accounts provide access to databases for applications or other services, often with restricted permissions based on specific needs.
- **Third-party service accounts:** Third-party service accounts are used to integrate with external services like payment gateways, email providers, or social media platforms.

System accounts

- **Root or administrator accounts:** These are the highest-privilege accounts on operating systems, allowing complete control over the system.
- **Service accounts for system processes:** Background processes like scheduled tasks, system updates, or logging services use these accounts.
- **Network service accounts:** These are used for network-related tasks like authentication, directory services, or file sharing.

Application accounts

- **Application-specific service accounts:** These accounts are created within an application to handle tasks like data processing, communication with external services, or scheduled jobs.
- **Embedded database accounts:** Many applications have built-in databases that use NHIs for managing access and data security.
- **Third-party integrations:** Applications might use NHIs to connect with other applications or services they depend on.

Why should companies care about non-human identity management?

→ **Exponential growth of NHIs:** With the rise of cloud computing, IoT, automation, and AI, the number of non-human entities needing access to digital resources has exploded. Non-human identity management is crucial for keeping track of these identities and managing their permissions effectively.

→ **Security imperative:** NHIs are increasingly targeted in cyberattacks, as they often have privileged access to sensitive data and systems. Without proper management, compromised NHIs can lead to devastating breaches. Non-human identity management allows organizations to proactively secure these identities, implement robust authentication and authorization mechanisms, and minimize the risk of unauthorized access.

→ **Operational efficiency**: Automation and cloud technologies rely on NHIs to interact and communicate seamlessly. Effective non-human identity management streamlines these processes, reducing manual intervention and human error. This leads to improved operational efficiency, faster application deployments, and better resource utilization.

→ **Regulatory requirements:** Several regulatory frameworks, like GDPR, HIPAA, and PCI DSS, mandate strict data protection and access controls. Non-human identity management helps organizations adhere to these standards by providing granular control over NHI access and generating audit trails for compliance reporting.

→ **Scalability and agility:** As organizations grow and embrace new technologies, the number of NHIs will continue to multiply. Non-human identity management provides a scalable and adaptable framework for managing these identities, allowing organizations to maintain control and security even in complex and dynamic environments.

How does non-human identity management work?

Non-human identity management involves authenticating, authorizing, and managing the lifecycle of non-human identities to ensure secure and efficient operations in today's technology-driven organizations. As the number and complexity of NHIs continue to grow, robust non-human identity management practices are essential for maintaining a strong security posture and mitigating risks.

Identity provisioning

→ NHIs are created either manually by administrators or automatically through scripts or configuration management tools.

→ Each NHI is assigned unique credentials, like API keys, tokens, or certificates.

→ NHIs are associated with metadata like purpose, permissions, and expiration dates.

Authentication and authorization

→ NHIs present their credentials when attempting to access resources.

→ The relevant system verifies the credentials and checks whether the NHI has the necessary permissions to perform the requested action.

→ Granular access controls are used to determine which resources and actions each NHI can access.

Lifecycle management

→ Continuous monitoring of NHI activity identifies suspicious behavior or unauthorized access attempts.

→ Regular rotation of credentials reduces the risk of compromise.

→ NHIs are promptly disabled or deleted when they are no longer needed or compromise is suspected.

Centralized management

→ A comprehensive inventory of all NHIs, their associated attributes, and their usage patterns is created and maintained.

→ Consistent policies are defined and enforced across all NHIs to ensure security and compliance.

→ NHI activity is recorded in detailed logs for compliance and security analysis.

Challenges in non-human identity management

→ **NHIs vastly outnumber human identities**: It's difficult to track, monitor, and secure each individual NHI in this sprawling landscape, increasing the potential for oversight and vulnerabilities.

→ **Limited visibility:** NHIs are created and managed across various systems and teams, leading to a lack of centralized visibility. This makes it challenging to understand the full scope of NHI activity, identify potential risks, and enforce consistent security policies.

→ **Overprovisioning and privilege creep:** NHIs are often granted excessive permissions, either intentionally for convenience or unintentionally due to a lack of understanding. This creates a significant security risk, as compromised NHIs can gain access to sensitive data and systems they shouldn't have.

→ **Credential management:** Securely storing and managing NHI credentials like API keys and tokens is critical. If these credentials are mishandled or fall into the wrong hands, they can be used to impersonate legitimate NHIs and gain unauthorized access.

→ **Lack of automation:** Manual NHI management processes are prone to errors, inconsistencies, and delays. Automation can streamline provisioning, deprovisioning, and monitoring, but implementing effective automation requires careful planning and integration.

→ **Evolving threat landscape:** Cybercriminals are constantly developing new techniques to exploit NHIs, making it essential to stay updated on the latest threats and adopt proactive security measures.

→ **Shadow IT:** The use of unauthorized or unmanaged NHIs, often created by developers or other teams outside of IT, can lead to significant security blind spots and increased risk.

→ **Lack of awareness and training:** Insufficient understanding of NHI security risks and best practices among developers, administrators, and other stakeholders can lead to misconfigurations and vulnerabilities.

Non-human identity management best practices

Embrace zero trust

- → Never assume trustworthiness. Always verify every access request, even if an NHI is already authenticated. This ensures that compromised or misconfigured identities can't bypass security controls.
- → Employ mechanisms for ongoing verification of NHIs throughout their lifecycle, not just during initial access requests.

Enforce the principle of least privilege

- → Assign permissions at the most granular level possible, allowing NHIs to access only the specific data and resources they require to function.
- → Store NHI credentials in secure vaults with encryption and multi-factor authentication (MFA) for access.
- → Embrace dynamic authorization. Implement systems that can adjust permissions based on context or risk factors, providing additional protection for sensitive data.

Monitor for behavioral anomalies

- → Implement advanced behavioral analytics to detect unusual NHI activity patterns, such as accessing data outside of normal hours or attempting to perform actions beyond their usual scope.
- → Configure real-time alerts for suspicious NHI behavior, enabling security teams to respond quickly and investigate potential threats.

Avoid over-permissioned identities

- → Conduct regular reviews of NHI permissions to identify and remove any excessive access rights.
- → Regularly rotate NHI credentials and promptly revoke access for NHIs that are no longer needed or potentially compromised.
- → Use privileged access management (PAM) solutions to manage and monitor NHIs with elevated privileges, ensuring accountability and reducing the risk of misuse.

Conduct regular audits

- → Schedule routine audits of NHI permissions to identify unused or excessive access, ensuring that privileges align with current needs and security policies.
- → Leverage automation to streamline the auditing process and reduce the risk of human error.

To learn how ConductorOne can help you streamline all your identity management, enforce least privilege, and achieve a zero-trust security model, talk to our team.