CYBERARK®
The Identity Security Company

EBOOK

Key Considerations for Securing Different Non-human Identities

Practical approaches to securing secrets used by seven types of non-human identities

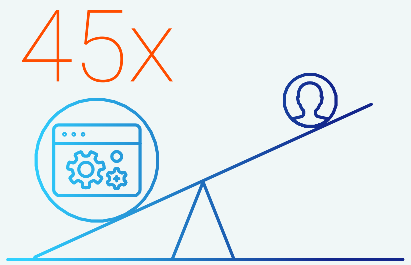
Table of Contents

Introduction

Security teams have spent years securing human identities with privileged access management (PAM) programs and ensuring the principle of least privilege is adhered to throughout the enterprise. But in recent years, organizations have experienced an explosion of non-human identities, thanks to large shifts like digital transformation efforts and the use of automation tools and processes. Today's enterprises are powered by a wide variety of machine identities, from microservices running in the cloud and unattended bots in robotic process automation (RPA) workloads to static or mainframe applications that still sit on-premises. These non-human identities rely on secrets (passwords, SSH keys, API keys and more) to access the critical systems needed to do their jobs.

As the number of non-human identities and secrets to protect grows exponentially, security teams must expand their PAM programs to secure all elements of privileged access — and that means protecting human ***and*** non-human identities — without slowing down development teams or delaying deployments. To keep up with the rapid pace of development, security and operations teams must work together to ensure that secrets management processes for non-human identities are scalable, automated, don't interfere with mission-critical applications' availability and are centralized to reduce the issue of islands of security and vault sprawl.

Acceleration of Non-human Identities



Machine identities outnumber human ones

Machine identities now outweigh human ones in your average organization by a factor of 45x.

CyberArk, "2022 Identity Security Threat Landscape," March 2022.

Non-human Identity Security Challenges Are on the Rise

The Impact of Digital Transformation Initiatives

While the digital transformation efforts of recent years have great productivity benefits, they can also impact security — especially when it comes to securing non-human identities. DevOps tools and processes have helped accelerate these digital transformation initiatives. Public cloud usage and cloud-native application development have continued to rise, allowing organizations to innovate faster than ever before. However, disjointed DevOps security policies can slow down development. This frustrates developers and leads to risky workarounds like hard coding credentials. Meanwhile, security teams find it challenging to keep up with the increasing number of non-human identities and secrets that need to be managed across diverse environments — from multiple public cloud and private cloud instances to on-premises infrastructure. A balance has to be struck between development velocity and security best practices to not only secure secrets used across the enterprise but to fold these efforts into a broader initiative of securing identities, both human and non-human.

85%

of respondents will leverage three or more public cloud providers in the next twelve months.[1]

40%

of respondents that used containers for most or all production applications and business segments listed security as their top concern.[2]

[1] Enterprise Strategy Group, "The Holistic Identity Security Model," February 2023.
[2] Red Hat, "2022 State of Kubernetes Security Report," 5 May 2022.

98%

of respondents feel that securing DevOps environments is critical or important to securing the software supply chain.[3]

55%

of security pros reported their organization experienced an incident or breach involving supply chain or third-party providers in the past 12 months.[4]

71%

of organizations surveyed by CyberArk have suffered a software supply chain attack.[5]

Increased Executive Risk Awareness

In recent years, there have been several high-profile breaches in software supply chain security. From **Solar Winds** and **Codecov** to **Uber** and **CircleCI**, digital supply chain attacks have impacted thousands of organizations and put software supply chain security on everyone's radar.

Additionally, a presidential executive order "charge[d] multiple agencies — including NIST — with enhancing cybersecurity through a variety of initiatives related to the security and integrity of the software supply chain."[6] This national attention and recent attacks have dramatically raised awareness of the need to secure software supply chains, from application development to the apps themselves. Attackers look for the weakest link in the supply chain to exploit. Even insignificant applications can expose organizations to attackers, as seen in the 2022 Uber breach. And the risk is growing as attackers shift left into the software development process and software supply chain. Increasingly boards and executives are asking security leaders how the enterprise's application portfolios and software supply chains are secured and how the organization is defending itself from attacks.

[3] Enterprise Strategy Group, "The Holistic Identity Security Model," February 2023.
[4] Forrester. "Predictions 2022: Cybersecurity, Risk, And Privacy." Jeff Pollard, October 28, 2021.
[5] CyberArk, "2022 Identity Security Threat Landscape," March 2022.
[6] NIST. "Improving the Nation's Cybersecurity: NIST's Responsibilities under the May 2021 Executive Order."

The Four Challenges of Secrets Management

Given these trends, security teams are seeing secrets management grow in both visibility and priority within the larger organization and at the executive level, particularly as organizations focus on securing all their identities, both human and non-human. But security teams face four challenges as they work to secure secrets across the enterprise.

1. **Wide range of non-human identities.** As we'll discuss in depth in the next section, there are a number of different areas where non-human identities operate within your enterprise. From cloud-native applications and Internet of Things (IoT) devices to Java and mainframe apps, there are a lot of non-human identities spread across your environments, all of which use secrets. Each of these different categories has its own unique challenges when it comes to secrets management.

2. **Secrets sprawl.** DevOps tools and cloud platforms often have built-in secrets management functions that developers may leverage to keep things moving quickly. However, this can lead to "secrets sprawl" or "vault sprawl," where secrets are scattered across multiple vaults owned by various project teams and production environments, creating a huge operational challenge.

3. **Managing at scale.** Businesses depend on machine identities to perform mission-critical tasks and operate reliably. Secrets management must be conducted at scale across all environments and tools for the business to maintain velocity.

4. **Meeting developers where they are.** Development teams are under pressure to innovate faster than ever, working in a variety of DevOps tools, containerization platforms and cloud service providers to meet their tight deadlines. Interruptions to their workflows or delays in deployment for security measures will not only cause frustration but can interfere with meeting business requirements. Instead, security teams should find secrets management solutions that meet developers where they are, providing native solutions and self-service capabilities that integrate seamlessly into their existing workflows.

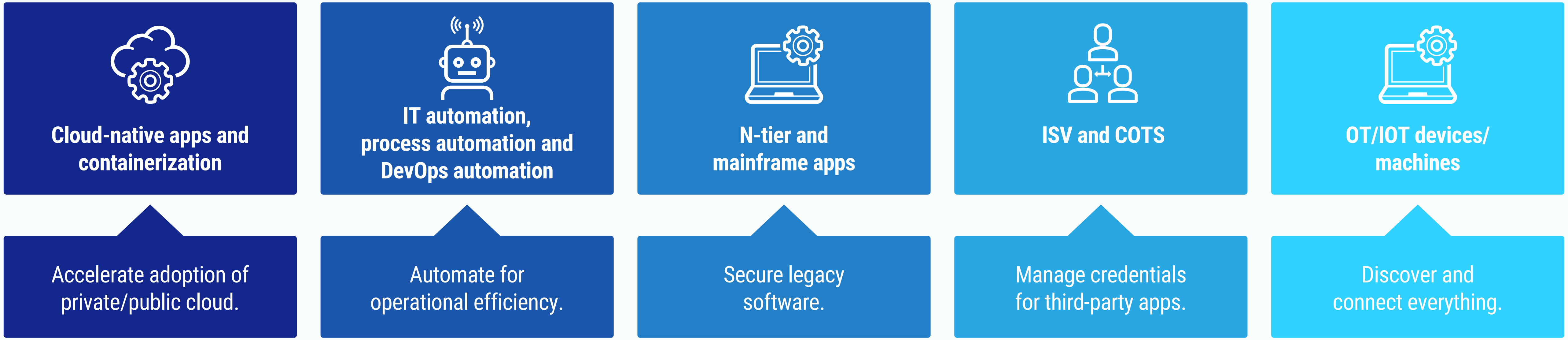
The Seven Types of Non-human Identities to Secure

There are likely several areas across your organization that house non-human identities using secrets that need to be managed and secured.

1. Cloud environments and cloud-native applications
2. DevOps tools, CI/CD pipelines and the software supply chain
3. Automation tools and scripts
4. RPA workloads
5. Commercial-off-the-shelf (COTS) and independent software (ISV) applications
6. N-tier/static homegrown applications
7. Mainframe applications

Each of these areas can have its own unique challenges, as well as different stakeholders who need to be engaged to ensure that the application identities are accessing secrets securely. And this is where that balancing act we mentioned at the beginning comes into play. Developers want easy-to-use solutions that don't slow them down. Security wants to make sure new applications are securely accessing resources. And operations leaders want to rapidly deploy new apps at scale, potentially across multiple cloud regions and multiple public clouds.

The Problem: CISOs Want Enterprise-wide Privilege Policies



Cloud-native apps and containerization

Accelerate adoption of private/public cloud.

IT automation, process automation and DevOps automation

Automate for operational efficiency.

N-tier and mainframe apps

Secure legacy software.

ISV and COTS

Manage credentials for third-party apps.

OT/IOT devices/ machines

Discover and connect everything.

1 Cloud Environments and Cloud-Native Apps

The cloud enables applications to scale dynamically to respond to business needs. Many organizations use multiple cloud service providers (CSPs) to maintain pricing control, enable flexibility and avoid cloud vendor lock-in. Each CSP has their own method for storing, accessing and managing secrets. The cloud-native applications built in these platforms are continually updated using CI/CD processes and often use secrets to communicate with other microservices in the cloud environment to run. Beyond that, developers also use containers — packages of software and all its necessary dependencies — so that they can automate manual work. These containers also have secrets, and hard-coded secrets or misconfigured container images have been a point of entry for attackers in the past.

Developers who are under pressure to deploy new features can potentially take shortcuts such as hard coding secrets or skipping security requirements to get applications out the door faster. They may also run into compliance roadblocks when it's time to push applications into production if they are unable to automatically rotate credentials or rely on unsupported open-source software for critical capabilities. But security processes that require code changes or changes to developer workflows will inevitably introduce delays and create frustration for cloud-native developers who need to move fast to meet business directives.

Tips for Securing Cloud Environments and Cloud-native Apps

To meet these challenges when working in the cloud, security teams should find a way to seamlessly manage secrets for cloud-native applications, meeting them where they are (i.e., the CSP they're working in), rather than trying to force a solution that slows them down or adds more work to their plates. Additionally, for organizations operating in multi-cloud or hybrid environments, a SaaS-based, centralized solution can help security teams manage secrets across multiple clouds and enable cloud portability for applications, so developers don't have to worry about code changes if they shift to a different cloud instance later on and security teams can work from a single pane of glass.

Working in the Cloud

Characteristics

- Dynamic apps running in containerized and cloud environments
- Continually updated using CI/CD processes
- Cloud environments with unique secret storage and management processes

Security Challenges

- Developers need to be able to work dynamically and at scale.
- Developers can take shortcuts or skip over security requirements.
- Compliance roadblocks can be created from not meeting corporate security requirements.
- Underlying DevOps tools and container platforms can lack security.
- Code repositories can expose secrets and cloud access keys.

DevOps Tools and CI/CD Pipelines

Characteristics

- Many different tools, usually selected by developers
- DevOps tools with high levels of privileged access
- May have built-in secrets management capabilities but cannot rotate or share secrets
- High levels of automation

Security Challenges

- Security has to shift left to be involved earlier in the development cycle.
- DevOps admins and developers may choose to use built-in secrets management functions, contributing to secrets or vault sprawl.
- Human checks and balances may need to be forced.
- An attacker may be able to escalate access once a tool is compromised ("Tier Zero").

2 DevOps Tools, CI/CD Pipelines and the Software Supply Chain

As mentioned previously, software supply chain attacks targeting continuous integration/continuous delivery (CI/CD) pipelines can be wide-ranging, affecting thousands of organizations downstream from the original point of attack. All it takes is one exposed credential on a public code repository or a misconfigured CI/CD tool for an attacker to compromise a supplier and thus their customers and partners. Unprotected secrets can expose an organization's entire IT environment, rippling downstream to their customers and partners.

DevOps tools typically require a high level of privileged access to perform their tasks. Thus, CI/CD pipelines and other DevOps tools are known as "Tier Zero" assets, meaning if an attacker gains access to these assets, they can then access more privileged credentials. Both the human identities (the developer endpoint) and the non-human identities within the pipeline need to be protected in order to mitigate this type of attack escalation (as seen in the CircleCI breach).

Tips for Software Supply Chain Security and Securing DevOps Tools

When it comes to securing DevOps tools, security needs to "shift left" to partner with developers and ensure that secrets aren't accidentally exposed, such as by removing hard-coded and default passwords or rotating passwords frequently and enabling multi-factor authentication — all without slowing down the pace of development. Additionally, because of the number of DevOps tools used, security should make developers and DevOps admins aware of the risk of vault sprawl (managing secrets using the built-in secrets management functions within the different tools) and make efforts to centralize secrets management.

3 Automation Tools and Scripts

Automation tools and scripts can be powerful and perform complex IT and other related tasks. But they can also be very simple, such as a basic PowerShell script used infrequently. Some automation tools and scripts will require high levels of privileged access to perform tasks such as performing updates across an entire compute infrastructure.

There are multiple challenges that need to be addressed to help ensure the security of the organization when it comes to automation scripts. Too often, these tools and scripts are overlooked when security teams search for potential vulnerabilities. But if the script has a valuable credential hard coded, it doesn't matter how basic or dated the script might be; a high-value embedded credential can be exposed to and leveraged by an attacker. This is exactly what occurred in the 2022 Uber exploit: a basic script with the credentials needed to administer the privileged access management (PAM) system was left in a file system that an attacker then accessed and stole the credentials from.

Additionally, these automation tools can be very powerful and need to be managed and privileged accordingly with policies such as just-in-time access, least privilege and audited access.

Tips for Automation Tools and Scripts

Security needs to aggressively work to prevent the use of hard-coded credentials in automation tools and scripts and establish policies that regularly rotate credentials so that if credentials are hard coded in scripts, they quickly become invalid. A big challenge with scripts and playbooks is that they are easily copied, replicated and shared, so control is quickly lost if a script includes an embedded credential. Someone can also easily post a copy to a repository.

At a basic level, all scripts and automation tools should obtain the necessary credentials from a centralized secrets management solution that gives security teams visibility into the tools using those credentials. The centralized solution can then enforce best practices and policies such as Zero Trust, just-in-time access, segregation of duties and, if needed, forced human approval with multi-factor authentication.

Automation Tools and Scripts

Characteristics

- Many different tools and scripts used by IT admins and other functions
- May require and be given high levels of privileges to complete their tasks
- May be run manually or fully automated

Security Challenges

- Too often scripts use embedded hard-coded credentials and can be posted to repositories.
- Scripts may be overlooked as a security vulnerability because of their simplicity.
- Ease of replicating and infrequent use make scripts hard to track.
- Some automation tools have built-in (native) secrets management capabilities that can lead to secrets sprawl and vault sprawl.
- Attackers can still exploit high-value credentials even if the tool is basic.

COTS and ISV Applications

Characteristics

- Typically owned by IT and security teams
- May require very high levels of privileged access
- Can sometimes access sensitive personal data

Security Challenges

- These apps require vendor-developed integrations.
- They are vulnerable to weaknesses in the vendor's software supply chain and CI/CD processes.
- High levels of access mean there is a high level of exposure if they are compromised by an attacker.
- Personal information stored in business applications could be exposed in a data breach.
- Least privilege and just-in-time access are imperative to reduce risk.

4 COTS and ISV Applications

Commercial-off-the-shelf (COTS) and independent software vendor (ISV) applications both require a high level of privileged access to do their jobs. A single organization may use a variety of these applications across their enterprise, but they typically fall into two categories:

- **Standard information technology (IT) security and management solutions –** These applications include critical security tools such as vulnerability scanners, IT operations applications (i.e., automated backup tools), inventory discovery software and identity governance and administration solutions. Due to the actions they need to perform and the systems they require access to, these tools have very high levels of privileged access that could be dangerous if exploited by an attacker.
- **Business applications –** Many business applications (such as ERP systems and human resources applications) have access to sensitive data, such as personally identifiable information (PII), personal health information (PHI) or cardholder information subject to Payment Card Industry (PCI) security requirements. Unauthorized access to privileged credentials used by these applications could allow an attacker to steal critical employee or customer information.

Tips for Securing COTS and ISV Applications

Because of the critical nature of these applications and their high levels of both privileged access and vulnerable data, it is imperative that security teams control access to these applications. Vaulting and rotating credentials is a key step, which can be accomplished with out-of-the-box integrations with a secrets management solution. Additionally, implementing just-in-time access can ensure these applications only have access to the data or systems they need to do their job at the point of time they need it, rather than standing access what could potentially be exploited by an attacker if a credential is compromised. For instance, when a vulnerability scanner needs to scan the network, it can pull the necessary credentials from a vault and use it for the time period needed for the scan. Then the vault can pull back that credential when the scan is complete. That way, if the vulnerability scanner is compromised, an attacker would not have standing access to valuable network resources.

RPA Processes

Characteristics

- Can be either attended by a human supervisor or unattended
- Require high levels of access to other business systems
- Growing rapidly

Security Challenges

- Manual rotation and management processes don't scale.
- Security can be seen as a blocker to deployment or operational efficiency requirements.
- Security wants easy-to-use integrations to minimize security issues and speed up deployment.

Securing the credentials used by bots is paramount, but doing so without delaying the deployment of new automated processes can be a challenge.

5 Robotic Process Automation (RPA)

Bots used in RPA processes require very high levels of privileges to do their jobs. Organizations that only deploy attended RPA solutions in a few instances can access the privileged credentials the bot needs to do its job from the human attendant — essentially a manual process that is secured in the same way that a privileged access management (PAM) solution secures human credentials. At scale, however, this immediately introduces operational challenges. Additionally, unattended bots don't have a human supervisor, so credentials must be handled differently. Manual rotations may be acceptable with small deployments, but these approaches do not scale.

According to a CyberArk survey, "Only 28% of organizations have identity security controls in place to secure bots/RPA. In fact, 74% say security concerns are slowing down RPA and bot deployments."[7] RPA usage is also spreading beyond IT to "citizen developers" within organizations, who create their own bots to automate simple, repetitive tasks. Citizen developers may not be as aware of bot security requirements. Additionally, once deployed, credentials for bots need to be rotated based on policy.

Tips for Securing RPA

Security teams need to ensure that they are enabling RPA velocity while also centrally managing policies to stay compliant and defend against attacks. Centralizing and automating secrets management for RPA bots can allow security teams to work at scale without delaying deployments. Credentials can be rotated automatically, and all bot credentials can be viewed from a single pane of glass. This gives security and operational teams better visibility, particularly over unattended bots.

[7] CyberArk, "2022 Identity Security Threat Landscape," March 2022.

6 N-Tier/Static Homegrown Applications

While many of the above applications leverage newer digital innovations such as the cloud and automation, most organizations still leverage a variety of internally developed applications hosted in corporate datacenters or in the cloud. These applications include a variety of traditional environments (such as Java) and operating systems, including Unix/Linux, which meet the needs of many of the enterprise's mission-critical applications necessary to run the business.

Static applications require privileged access to various IT systems and datastores. They typically operate on an agent-based approach, in which a local copy of the credential is stored near the application to avoid network issues and failover. Just as with dynamic, cloud-native apps, developers often hard code secrets into these applications, providing an opening for attackers. These homegrown applications commonly have overprivileged permissions, a remnant of a time when some developers requested all permissions rather than tailoring the appropriate access rights. In many cases, the developers or contractors originally responsible for the application are long gone. Additionally, secrets for some of these applications cannot be automatically rotated.

Tips for Securing N-tier/Static Homegrown Applications

For these applications, security teams must rotate secrets based on policy while keeping these critical applications running to meet high availability and business continuity requirements. Discovery and removal of any hard-coded credentials are key, as is vaulting the secrets in a central location for greater visibility. Out-of-the-box integrations with a centralized secrets management solution can make it easier for security teams to manage permissions and right-size access rights.

N-Tier/Static Homegrown Applications

Characteristics

- Often mission-critical applications
- Written in various languages, including Java variants and the widely used .NET
- Housed in on-premises environments

Security Challenges

- Credentials are hard coded or locally stored, introducing risk if compromised.
- Automatic rotation is sometimes not possible for the credentials used by these apps.
- Access rights need to be better tailored, as these apps can be over-permissioned.
- They need to easily connect to other systems and applications.

Static applications require privileged access to various IT systems and datastores.

Mainframes

Characteristics

- Mission-critical applications
- Handle high transaction volume
- Have dual password structure

Security Challenges

- Credential rotation can potentially interrupt high-volume transactions.
- Credentials are hard coded or locally stored, introducing risk if compromised.
- High levels of reliability are required.

> Think of a credit card processing application, which processes thousands of transactions a second. Even a moment's interruption could be costly for an enterprise.

7 Mainframe Applications

Like N-tier applications, applications hosted on mainframes (such as zOS) are still widely used by enterprises for specific use cases. These are the most mission-critical applications an enterprise has, and it's vital that these applications do not experience outages or have their processes interrupted by security procedures. Think of a credit card processing application, which processes thousands of transactions a second. Even a moment's interruption could be costly for an enterprise.

Because of that, one of the biggest challenges with applications handling high transaction volumes, which are frequently hosted on mainframes, is credential rotation. Because these applications manage high transactional volumes, security teams must ensure that credential rotation does not result in dropping any transactions as credentials are being rotated. Mainframe applications often use a dual password structure to prevent any transaction from being dropped during rotation. For example, an application will access IT resources using a primary credential. Then, when rotation is required, the application switches over to using a secondary credential, which becomes the new primary credential, while the new password being rotated in becomes the new secondary credential. The other challenge with these types of applications is that to meet high availability requirements, credentials are often locally cached in case the application cannot fetch a credential from the secrets manager. Finally, as with N-tier applications, developers also often hard code secrets into mainframe applications, and these applications commonly have high levels of privileged access to perform their tasks.

While these rotation approaches can also be found with other application types, they are essential for mission-critical, high-volume applications running on mainframes. For these applications, security teams must manage secrets carefully while making sure not to interrupt important tasks and processes.

Tips for Securing Mainframe Applications

For these identities, security teams must manage secrets carefully while making sure not to interrupt important tasks and processes. Automated rotation with a secrets management tool is possible, while adhering to the dual password structure. Just as with N-tier/static homegrown applications, security teams should ensure no secrets are hard coded or locally stored and that they have the right amount of access for the tasks they need to accomplish.

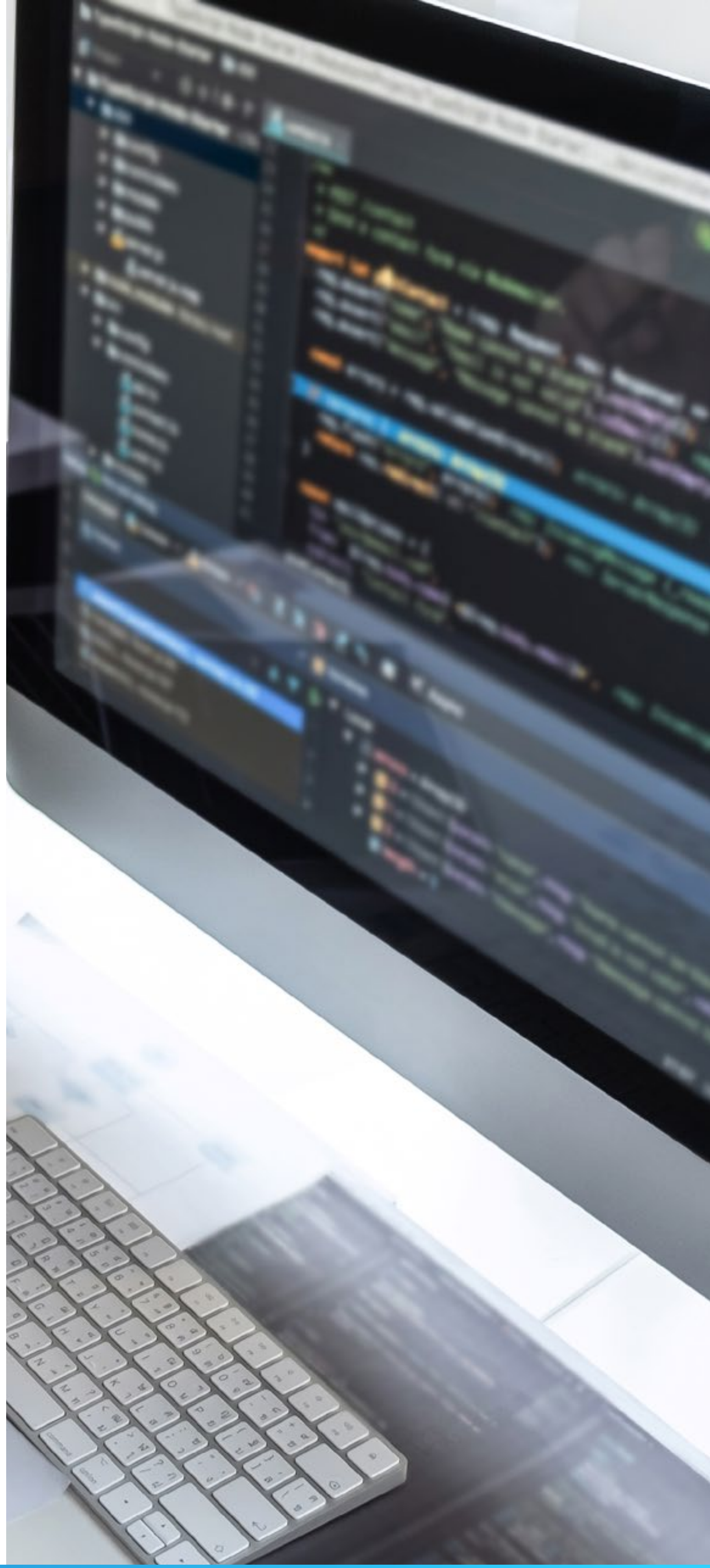
A Centralized Approach for Securing All Non-human Identities

Security teams today are facing a tough challenge when it comes to securing all non-human identities and secrets across their enterprise. There are not only types of non-human identities, each with their own unique challenges and characteristics. There are also multiple stakeholders with varying priorities involved in securing each of these different types of identities — such as native application developers, RPA automation specialists, IT and operations. Without the right approach, the need to secure secrets for all identities can overload security teams. All of these secrets need to be rotated, secured, audited and removed when no longer needed — across the entire enterprise landscape.

As the number of identities grows, so does the risk of operational challenges such as secrets or vault sprawl and the ways in which secrets can be exposed — whether that be via the cloud, an endpoint or an attacker breach of an application with critical privileges. At the same time, security must ensure that applications continue to run at scale with a high degree of reliability to ensure that operational efficiency is not impeded and that the digital business is enabled. Security teams also require more automation to keep up with the growing number and breadth of non-human identities spread across the enterprise to ensure that compliance needs are met.

Increasingly, large enterprises are taking a centralized approach to managing, rotating and auditing secrets across their environments. Centralizing secrets management:

- Reduces "islands of security."
- Gives the security team visibility across the entire organization.
- Enables developers and application owners to limit identities to the credentials they need to do their job.
- Allows organizations to move at a fast pace without sacrificing security.
- Eliminates code changes and simplifies secrets management for developers through out-of-the-box integrations, native tool capabilities and self-service solutions.
- Provides centralized audit logs so that security teams can quickly see which identities have fetched secrets and which had access to what resource.



Building a More Effective Secrets Management Program

But you can't build a secrets management program in a day. As we've seen with all the types of non-human identities, there is a lot of complexity that security teams have to navigate in order to secure secrets across the entire enterprise. Instead, think of secrets management as a journey, one in which you're building a foundation block by block that can scale to the next type of non-human identity as you need it to.

You may have already started securing your secrets, but as you look to build out your program, think about the ideal end state for your security team: knowing where all your non-human identities are and centrally securing them. Then work backward from there. What are the most critical identities to secure first? The easiest? Which ones will require working with additional stakeholders and thus may require more time or thought?

At CyberArk, what we've seen work best for our customers when it comes to building toward this goal is a phased approach. Each of these steps will get you closer to centrally managing non-human identities and the secrets they use.

Phase 1 - Start With the Essentials

The big lesson here is that you have to start somewhere. Even if you already have some secrets management in place, looking at all the places where you have non-human identities that still need securing can feel overwhelming. But you don’t have to secure them all at once. Even a quick analysis and some fast fixes can get you started, and then you can build out your program from there. A good place to start is familiarizing yourself with what your DevOps teams are doing. Start a conversation with them, including introducing some security best practices, so that you can better assess what’s going on.

One common theme you likely picked up on across the different identity types was the need to find and remove hard-coded secrets. They can be a critical security vulnerability and so are a great place to start your secrets management journey. And don't make the mistake of thinking that just encrypting the secret at rest will solve the issue. It just creates more technical debt for down the road when you have to rotate that secret. In this first phase, you want to set yourself up to expand later with a scalable solution that can grow in a modular way as your secrets management program matures.

If you're a PAM expert or owner and have a SaaS-based secrets manager, one approach for building your own experience and showing developers how to secure credentials is to use the secrets manager to secure the credentials used by your PAM automation scripts.

Phase 2 - Accelerate Your Program

Once you've gotten the essentials taken care of by securing some secrets and non-human identities, it's time to look to how you want to expand outward to eventually cover all types of non-human identities. As we've mentioned, this is not a one-step process and will likely take time and multiple iterations. And each organization's needs will be different.

To determine how to expand your secrets management program beyond the essentials, you can leverage the CyberArk Blueprint. The CyberArk Blueprint for Identity Security Success lays out a prescriptive, risk-aligned approach for establishing and maintaining an effective identity security strategy — including a robust, unified secrets management program for your entire enterprise. To learn more about the stages of the CyberArk Blueprint for secrets management, check out our eBook "The CISO Mandate: Accelerate Securing All Application Identities."

Phase 3 - Expand as You Grow

The good news about a scalable solution that grows with you is that you can use it to secure secrets for specific business technologies as you start using them. This includes things like RPA, IoT and cloud-native environments. A solution that scales to fit your needs means you don't have to worry about finding another vendor or tool to secure secrets for these tools down the road as your business continues to grow and invest in innovative technology.

Five Tips for Securing Non-human Identities

Here are five tips to help you get started in securing non-human identities and the secrets they use across your enterprise.

1. **Integrate secrets management with existing tools and applications.** As we covered, there are a lot of tools out there used today that access secrets. Integrating a solution directly with these various tools helps simplify secrets management for security applications, as well as the tools that are used to build apps and the containerized environments that the apps run on.

2. **Centralize your secrets management and reduce secrets sprawl.** Instead of juggling secrets management policies spread out across different tools and teams, a centralized platform means you only have one program to support, with the wide visibility you need. A centralized solution also reduces the potential for unauthorized application access and associated implications of data theft and compromise, plus provides a single place for audit and logging.

3. **Automate security functions to improve operational efficiency.** Automation means that tasks like rotation, audit and data collection can run in the background, rather than where they might disrupt workflows.

4. **Make security easy for developers — and provide easy-to-use options.** Enable your application teams by giving them the tools to maintain their velocity, rather than slow down their deployments. Developers want the path of least resistance, and self-service tools that integrate seamlessly with the solutions they're already using can ensure that security requirements are being met while not frustrating developers. Ideally, security teams can offer developers secrets management tools that "meet developers where they are" and avoid changes to the developers' experience and workflows.

5. **Prioritize the applications to secure.** Many CISOs want to secure all application types, but it is a process. Security teams must pick where to start and then build off that. When creating your secrets management roadmap, you'll need to figure out which applications to tackle and in which order. You can rank applications based on perceived business risk, vulnerabilities discovered during red team exercises or in response to known breaches and security incidents. You can also consider compliance risks and audit findings, prioritize pet projects for executives or factor in other internal priorities.

Next Steps

With a centralized secrets management solution, organizations see the following benefits:

1. They can **get visibility and control** of the secrets and other non-human credentials across the entire enterprise.
2. They can **deliver measurable cyber risk reduction** by removing islands of security and securing secrets across all application types.
3. They can **enable operational efficiency** by accelerating digital business initiatives.
4. They can **secure digital transformation** by automating security processes like secrets management.
5. And they can **satisfy audit and compliance** processes quicker and more easily than before.

Security teams empowered with a unified secrets management solution can secure the secrets used by all application types while giving developers the tools and native capabilities to avoid slowing down development and improving operational efficiency.

Ready to get started on your secrets management journey? Set up a 30-minute initial discussion to see how CyberArk SMEs can help you and your organization's planning process for securing secrets for almost any type of non-human identity.

Free Registration for SaaS Secrets Management Solutions

CyberArk is the global leader in Identity Security. Centered on privileged access management, CyberArk provides the most comprehensive security offering for any identity — human or machine — across business applications, distributed workforces, hybrid cloud workloads and throughout the DevOps lifecycle. The world's leading organizations trust CyberArk to help secure their most critical assets. To learn more about CyberArk, visit www.cyberark.com, read the CyberArk blogs or follow us on Twitter via @CyberArk, LinkedIn or Facebook.

CyberArk believes the information in this document is accurate as of its publication date. The information is provided without any express, statutory, or implied warranties and is subject to change without notice.
THIS PUBLICATION IS FOR INFORMATIONAL PURPOSES ONLY AND IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER WHETHER EXPRESSED OR IMPLIED, INCLUDING WARRANTY OF MERCHANTABILITY, FITNESS FOR ANY PARTICULAR PURPOSE, NON-INFRINGEMENT OR OTHERWISE. IN NO EVENT SHALL CYBERARK BE LIABLE FOR ANY DAMAGES WHATSOEVER, AND IN PARTICULAR CYBERARK SHALL NOT BE LIABLE FOR DIRECT, SPECIAL, INDIRECT, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, OR DAMAGES FOR LOST PROFITS, LOSS OF REVENUE OR LOSS OF USE, COST OF REPLACEMENT GOODS, LOSS OR DAMAGE TO DATA ARISING FROM USE OF OR IN RELIANCE ON THIS PUBLICATION, EVEN IF CYBERARK HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
U.S., 04.23 Doc: TSK-3738 (TSK-3180, TSK-997-A)

CYBERARK®
The Identity Security Company