SECURING ACCESS FOR AGILE DEVOPS TEAMS: PRIVILEGED ACCOUNT MANAGEMENT (PAM) AS CODE

INTRODUCTION: THE REVOLUTION IN DEVOPS

In the past, IT departments would typically segregate their software development team from their IT Operations teams. Developers who built the products that were being sold or used internally were kept separate from those that were deploying and supporting the applications. Often, the lack of contact and collaboration between the two groups would lead to frustration and problems when attempting to bring new code into production. This resulted in longer deployment cycles, slower innovation, and a greater strain on the IT Operations team when it came time to push new product features and applications to production.

These days, most organizations are rapidly adopting Agile methods of development that require Development and Operations work together as a DevOps team. DevOps is now characterized by its automation of infrastructure and workflows, and continuous measurement of application performance.

As a result:

- Development and IT Operations now work more closely together to remove bottlenecks in the software development life cycle (SDLC).
- Code is written, tested, and deployed in smaller chunks for incremental changes. Team members integrate work multiple times each day. A workflow process of Continuous Delivery, or Continuous Deployment, allows for micro releases on a continuous basis.
- Automation is fundamental, from testing code to provisioning computing resources. DevOps teams can quickly build products and features compatible with different sized environments. In doing so they can utilize dozens, hundreds, or thousands of servers using different types of hardware at multiple locations, including cloud, multi-cloud, and hybrid-cloud.
- High-performing, high velocity IT organizations enable DevOps to achieve 46x more frequent software deployments, 96x faster recovery from failures, and 440x faster lead time for changes.

By 2021, DevSecOps practices will be embedded in 80% of rapid development teams, up from 15% in 2017. - Gartner

ACCELERATING DEMANDS ON DEVOPS CREDENTIALING AND PROVISIONING

To keep pace with these trends, DevOps teams need better ways to quickly and securely provision new services and infrastructures without adding additional time and effort to their process. As they incorporate Agile methodologies, continuous iterations of codes and applications require constant work, revisions, updates, and new deployments.

In DevOps standard practices, infrastructure and services are often represented as code or configuration files to enable automation and provisioning. Each of these services and infrastructure supports are accessed with some sort of privileged account. Additionally, many services, VMs, and applications that are provisioned and deployed require rapid and secure access to different components, such as:

- An API
- Another Service
- A Service Account
- A Database
- An SSH Key

And more…

Many of these connections require the use of a hardcoded key or credential somewhere in the application's environment. However, hardcoded passwords of any type are a potential vulnerability that opens a backdoor into sensitive systems. Since these services are often expressed in configuration files, the privileged credential is often hardcoded in a file.

Security was named the #1 DevOps obstacle by 28% of enterprises in 2015 - ZDNet[1]

CHALLENGES OF ENSURING SECURE ACCESS FOR DEVOPS TEAMS

To establish and maintain best cybersecurity practices, IT departments face several challenges with DevOps environments.

- Protecting their DevOps environment and tools
- Discovering and removing hardcoded passwords from code
- Providing unique accounts and credentials to containers and services
- Managing both Development access and IT Operations access from a single location
- Gaining visibility into and between both development and IT operations
- Securing DevOps environments

Most important, teams must secure their DevOps environments and tools without adding extra work or hindering the Agile environment. Every DevOps platform has administrative privileged accounts that manage infrastructures, and these accounts must be discovered, protected, controlled, and managed. Any compromise of these master accounts will easily allow a hacker or malicious insider to override security controls designed to safeguard application development and deployment.

[1] http://www.zdnet.com/article/devops-delivers-survey-says/

SECURING DEVOPS APPLICATIONS

One of the primary vulnerabilities in application code stems from hardcoded passwords and keys. If someone gets access to hardcoded passwords, they can easily leverage them to gain access to the production environment and move laterally throughout the network undetected.

Because applications require access to multiple systems, and operate much like anyone using privileged accounts, the activity of that application must be monitored. While most DevOps teams are concerned about an application's process performance, how fast it performs, etc., they should also be able to monitor and alert on unusual behavior around the use of accounts for accessing systems and services. If malicious code is injected into an application, or an application account is hijacked, DevOps teams need a way to instantly detect and alert when an application has gone rogue.

Because security has typically operated as a separate functional team, maintaining proper controls during the DevOps process can impede or slow the development process. Security teams, however, are integrating automated tools for vulnerability scanning and analysis within the DevOps toolchain to help integrate security requirements. They are adapting security to the evolving DevOps process, breaking security out of its traditional silo, and engaging with DevOps teams to help safeguard the organization.

CENTRALLY CONTROLLING ACCESS: THE KEY TO A MORE SECURE DEVOPS ENVIRONMENT

Because Development and IT Operations staff work across a dynamic environment, it's important to have a central place to authorize, authenticate, and audit all access across multiple sessions. IT Departments need an automated, centrally managed system that can provide a simple way to rapidly provide the right access to the right people and the right systems. Privileged Access Management (PAM) software provides a proven way to securely manage credentials for DevOps teams without impacting their productivity. 60 to 80 percent of all security breaches involve the compromise of user and privileged account passwords, but traditional ways of identifying and managing privileged accounts---including those used by DevOps teams---often rely on manual, time-consuming tasks performed on an infrequent or ad-hoc basis.

Common passwords are shared without authorization across multiple systems, or default passwords are never changed — making them prime targets for attack. Once any user account is compromised, hackers often seek to take over privileged accounts to escalate their access to applications, data, and key administrative functions. Once gaining access to privileged account credentials, hackers can easily conceal their activities in the guise of a legitimate administrative user.

While Privileged Access Management (PAM) is typically established and run by enterprise IT Operations or Security departments, the benefits of PAM security can be readily extended to the DevOps team to help centrally manage and control privileged account credentials. By deploying a PAM solution across the entire enterprise, IT executives are better able to discover, control, protect, and manage these privileged accounts across all systems.

As the adoption of the DevOps processes accelerates, IT organizations are incorporating and automating traditional security functions to produce a DevSecOps model that allows the development team to continue working quickly---but securely---within an Agile framework. That means teams can securely write code and access appropriate systems whether those systems are on premise or in the cloud.

EXTENDING THE PROTECTIONS OF PAM SECURITY TO DEVOPS

When, for example, the IT Operations or Security department implements Privileged Access Management and moves toward "Least Privilege" access policies, they need solutions that will integrate with Agile development processes, allowing DevOps teams to operate as effectively and efficiently as possible.

Thycotic Secret Server offers a simple, easy to manage and effective PAM solution used by more than 7,500 organizations worldwide, including Fortune 500 companies. Thycotic Secret Server delivers the extensibility and customization ability that enterprise IT departments typically can't get from other PAM offerings. When IT Operations and Security teams implement Thycotic Secret Server to protect their privileged accounts for IT and business users, they can readily extend the same protective functionality to their DevOps team through the Secret Server SDK.

Thus, Secret Server SDK protects protects privileged credentials used in the DevOps environment with the same security best practices and management tools used throughout the rest of the enterprise, while maintaining the speed and scale demanded by today's DevOps teams.

THYCOTIC SOLUTIONS - PRIVILEGED ACCOUNT MANAGEMENT (PAM) AS CODE

Secret Server SDK provides development teams with the ability to utilize Thycotic's Secret Server PAM security software to quickly and rapidly leverage our PAM solution using code. Because developers are comfortable with code and creating scripts, Thycotic continues to ensure that its API allows them to do so comfortably, safely, and quickly.

One of the major challenges development teams face is the ability to replace hardcoded passwords and protect privileged accounts. Thycotic's Secret Server SDK functionality allows teams to remove hardcoded credentials and replace them with tokens. In addition, Thycotic's Privileged Behavior Analytics solution enables the DevOps team to monitor application accounts leveraging the tokens and detect anomalous or unusual behavior in near real time. This can provide an early warning of compromised behavior or an application gone rogue and prevent unauthorized escalation of privileges.

By extending Privileged Access Management security to the DevOps team, Thycotic Secret Server provides an additional layer of security for the enterprise without impacting the crucial workflows and efficiency of application development essential to business growth. CIOs and CISOs gain several advantages from the Secret Server SDK solution including:

- Now security teams can ensure the same level of privileged account protection for applications, scripts, and infrastructure used within an SDLC as they have for users logging into workstations, servers, and business applications.
- Secret Server SDK allows DevOps teams to avoid hardcoding secrets (credentials, keys, certificates, tokens) in scripts and using insecure repositories where those secrets can be hijacked and exploited.
- Using our extensible API, DevOps teams can easily integrate Secret Server SDK to connect every tool in the DevOps toolchain. Secret Server SDK scales for dynamic, rapidly changing needs with maximum resiliency. DevOps teams will be able to secure credentials for as many new environments as needed, on any operating system, whenever they need them.

The table here summarizes how Thycotic Secret Server and Secret Server SDK help to meet the challenges of keeping fast-moving DevOps teams safe and successful.

SECURITY CHALLENGES IN THE DEVOPS WORKFLOW: HOW THYCOTIC CAN HELP

DEVOPS WORKFLOW SECURITY CHALLENGES	HOW THYCOTIC HELPS MINIMIZE VULNERABILITIES WITHOUT IMPACTING WORKFLOW
Security vulnerabilities can easily multiply when developers embed hardcoded keys or credentials in a file within an application's environment. Privileged credentials can be exposed during build when developers store credentials in a repository such as GitHub, forget about them, and then commit them to production.	Secret Server SDK enables DevOps teams to avoid hardcoding secrets (credentials, keys, certificates, tokens, etc) in scripts, or in using insecure repositories where they can be hijacked and exploited.
DevOps teams often use open source code within the production process. Grabbing and integrating code found in a repository like GitHub could mean integrating hardcoded or shared credentials into your application.	With Thycotic, DevOps teams can discover and remove any existing hardcoded credentials and keys from applications, platform control and configuration scripts and replace them with tokens for increased security.
In a DevOps team, where many people need on-demand access, team members often share private keys and credentials for immediate access.	Secret Server removes the risks associated with humans managing privileged credentials. Automated management includes role-based access control, and SSH key rotation for service accounts to avoid weak or shared passwords. All credentials are checked in and stored safely in a secure vault. Usage is logged, providing a detailed recorded history of credential usage across any number of machines.

DEVOPS WORKFLOW SECURITY CHALLENGES	HOW THYCOTIC HELPS MINIMIZE VULNERABILITIES WITHOUT IMPACTING WORKFLOW
In an “infrastructure as code” model, the code itself acts as a privileged user. Administrative privileges are used by the configuration management and orchestration systems that continually spin up new servers, install software and make configuration changes throughout the SDLC.	Secret Server allows for infrastructure to be a privileged user, providing access across different tools and information pathways used by DevOps teams. Secret Server can also scale to manage the large number of secrets needed for resources in a constantly changing DevOps environment.
If malicious code gets injected into an application, or an application account is hijacked, DevOps teams need a way to instantly detect and alert when an application has gone rogue and prevent unauthorized escalation of privileges.	Secret Server logs all events and retains records to provide auditable assurance that privilege vulnerabilities have been eliminated from the software. If there is any unauthorized escalation of privileges, Thycotic helps you raise the fences to avoid any further damage.
Though most operations are automated, there are times when individual developers and Sys Admins need direct access to a problematic system to determine the root cause for a recurring issue and debugging.	Secret Server provides role-based access control and can provision temporary, one-time passwords or SSH keys, and enable SSH key rotation for service accounts.

CYBERSECURITY TEAM ADVANTAGES WITH SECRET SERVER

- Extending enterprise-grade privilege to the DevOps environment means security teams can ensure the same level of privileged account protection for applications, scripts, and infrastructure used within an SDLC as they rely on users logging into workstations, servers, and business applications.
- Policies and credential usage can be more easily managed across different teams and functions, all from a single tool with a unified view, enhancing control, and minimizing costs.
- Usage logging and behavior monitoring allow security analysis to detect any anomalous or unusual behavior to detect and respond to threats faster, preventing escalation of privileges.

DEVOPS TEAM ADVANTAGES WITH SECRET SERVER

- No need to set up or learn a new tool when adding Privilege Account Management to DevOps.
- Developers get the privileges they need to code, integrate, configure, build, test, verify, deploy and manage applications – all without relying on hardcoded or shared passwords that can be exploited.
- Secret Server SDK enables team members to push and pull credentials out of the Secret Server vault for any DevOps tools that need to be integrated.
- Building tools in-house, or downloading open source free tools to manage DevOps credential usage means additional coding headaches, isolated systems that don’t integrate with existing security or compliance policies and tools. In many cases free or in-house tools may exhibit weaker security controls than proven Privilege Access Management software.

OPERATIONS TEAMS

- Secret Server SDK readily scales to meet dynamic, rapidly changing needs. Credentials can be secured wherever and whenever - for as many new environments as you have, on any operating system.
- Secret Server SDK caches files so the high volume of calls from applications to databases and other applications happens at lightning speed – 10x faster than calling a web service.
- For maximum resiliency and guaranteed uptime, Secret Server SDK takes the credentials it has access to and stores them in the application itself. Even if Secret Server goes down, the application and environment keep working.
- Logging and monitoring insights provide assurance that the operating environment is healthy and secure.

Bottom Line: Secret Server SDK helps you protect privileges used in the DevOps environment with the same security best practices and management tools that Thycotic delivers across the enterprise.

This enables you to protect privileged accounts, implement least privilege policies, and meet compliance requirements while at the same time maintaining the speed and scale DevOps teams demand.

To learn more about how Thycotic Privileged Account Management solutions can help you secure your DevOps environment without impacting productivity, visit our website at thycotic.com/pam-devops/