Secret Server SDK
Extends Thycotic's Proven Privileged Access Protection to DevOps

THE CHALLENGE

To keep pace with constant demands to innovate, DevOps teams need 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 code and applications requires constant work, revisions, updates, and new deployments.

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

Organizations 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 DevOps Obstacle

Security was named the #1 DevOps obstacle by 28% of enterprises in 2015

- CA Technologies, *"DevOps: The Worst-Kept Secret to Winning in the Application Economy" (ca.com)*

THE SOLUTION

Secret Server SDK provides development teams with the ability to utilize Thycotic's Privileged Account Management solution within their code and configuration files. Because developers need to easily incorporate credentials for code and deployment scripts, Thycotic provides a command line interface to quickly link Secret Server's critical Privileged Account protections with the fast deployment process of a DevOps environment.

Thycotic's Secret Server SDK allows teams to remove hardcoded credentials from code, build scripts, and configuration files. By extending Privileged Access Management security to the DevOps team, the Secret Server SDK provides an additional layer of security for the enterprise without impacting the crucial workflows and efficiency of application development essential to business growth.

- Removes hardcoded passwords from code.
- Provides unique accounts and credentials to containers and services.
- Avoids using insecure repositories where those secrets can be hijacked and exploited.
- Can connect PAM security with tools used in your DevOps ecosystem.
- Scales on demand to meet dynamic, rapidly changing needs with maximum resiliency.

In addition, Thycotic's Privileged Behavior Analytics solution enables the DevOps team to monitor application accounts leveraging 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.

e: sales@thycotic.com
t: @thycotic
www.thycotic.com

SECRET SERVER SDK FEATURES

Tokens - Tokenized credentials in scripts and .NET web applications.

Key generation- Access temporary and permanent key generation via scripting.

Caching - 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.

Local encryption - Local encryption storage every place you install it. Secret Server takes the credentials it has access to and stores it in the application itself. That way, even if secret server goes down, the tool can keep working.

Auditing - Secret Server audits requests made by the SDK, providing an audit trail of credential usage.

SECRET SERVER SDK BENEFITS

IT Operations and Development teams gain several advantages from the Secret Server SDK solution including:

- 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 SDK and its extensible API, DevOps teams can easily integrate Secret Server into their DevOps system. 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, whenever they need them.

Licensing Requirements

Requires a paid edition of Secret Server and either Sampler Pack or API licenses.

System Requirements

- win10-x64 (Windows 10)
- centos-7-x64 (CentOS 7)
- rhel.7-x64 (Red Hat Enterprise 7)
- ubuntu.16.10-x64 (Ubuntu 16.10)
- osx.10.12-x64 (Mac OS 10.12)

ADVANTAGES ACROSS THE ENTERPRISE

For Cybersecurity Teams

- Extend enterprise-grade privileged account protection to DevOps.
- Effectively manage policies and credential usage from a single tool with unified view.
- Enable usage logging and behavior monitoring to detect and respond to threats faster, preventing escalation of privileges.
- Demonstrate compliance with automated reports.

For Development Teams

- No need to set up or learn a new tool to ensure privilege account protection.
- Get privileged credentials on demand to code, configure, build, test, verify, deploy, and manage applications – without relying on risky hardcoded or shared passwords.
- Extend credential protection to any tool requiring credentials in your DevOps environment.
- Avoid building credentialing tools in-house, or downloading open source free tools associated with increased risks and vulnerabilities.

For Operations Teams

- Readily scale to meet rapidly changing needs, securing credentials for new environments, anytime, anywhere.
- Cache credentials so that high volume calls, from applications to databases and other applications, occur at lightning speed – 10x faster than calling a web service.
- Deliver maximum resiliency and guaranteed uptime, with credentials stored in the application itself.