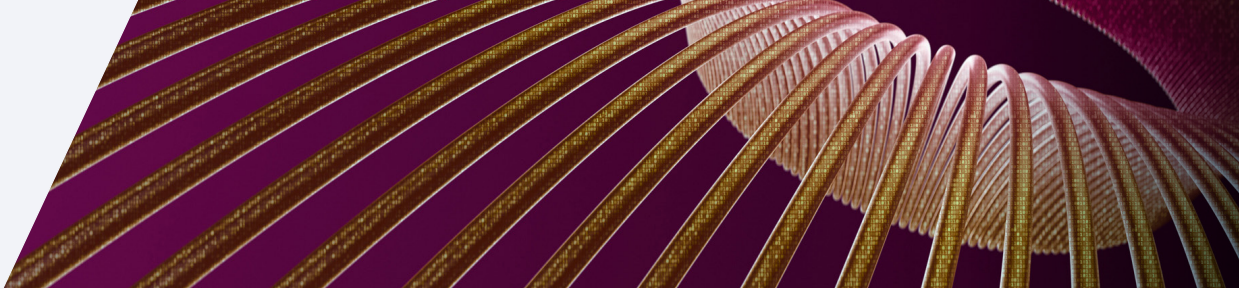
Venafi®



SOLUTION BRIEF

How InfoSec Can Secure the Code Signing Process

Addressing the problems created by poor code signing policies and processes

Who should read this: InfoSec leaders, CISOs and those responsible for protecting their company's software assets.

Enterprise executives know code signing is an important security control, for both self-defense and external reputation protection. Most organizations, however, overlook securing critical aspects of the process by which code is signed such as who is authorized to sign it, who needs to approve it, and what certificates are authorized to be used. This leaves businesses vulnerable to security and brand risks.

When a company's code is signed with a certificate, it is supposed to confirm the company is the author and the code is trustworthy, assuring the software's integrity. The company's customers expect products signed with its code signing certificates to be secure, and they must be able to rely on the brand to deliver safe products. But what happens when a legitimate code signing certificate is stolen?

Security Risk from Poor Code Signing Practices

Cybercriminals steal code signing private keys from legitimate companies to sign their malicious code. When signed with a private key and certificate from a legitimate company, the malware bears the identity of that company. This keeps the malicious code from triggering warnings when installed by unsuspecting users who believe the application is safe to install and use.

Code signing certificates can be used to sign any code, and that code can be time-stamped to effectively never expire. As a result, malware signed with stolen credentials can have a lasting impact on businesses, causing lost revenue, lost customer trust, damage to brand reputation and even legal liability.

This isn't a failure of code signing as a security control but a failure to protect code signing processes. While code signing involves critical security assets, such as private keys and certificates, poor code signing practices are often applied to securing these assets. Code signing is frequently performed and managed by developers, not InfoSec teams. Therefore, even though InfoSec teams are responsible for corporate information security, they may not have control over or visibility into code signing processes.

The Truth about Software Development Teams

Responsibility for code signing activities usually falls to each project's software development team. Companies can have hundreds to tens of thousands of geographically distributed developers who build different types of software, use different software methodologies and leverage different software tools. These differences make it extremely difficult for a corporate InfoSec team to have visibility into and control over all corporate code signing activities.

To make matters worse, InfoSec teams often have limited resources and possibly even limited influence. But perhaps more concerning, software teams often do not have PKI expertise, leading to unsafe practices, such as not securing private code signing keys.

Lack of Global Visibility

If malware discovered on the internet is signed with a legitimate code signing certificate from your company, would you know where to start looking to find the breach? If you were asked to comply with an audit of all corporate code signing activities, could you do that?

The InfoSec team needs to have enterprisewide visibility into all code signing certificates in use, not just those used for publicly signed code. In addition, visibility needs to include information on all code that has been signed, including who signed it, who approved it and what tools were used to perform the signing. But in code signing processes, these important elements are often left unmanaged and untracked.

Unprotected Private Keys

Development teams not securing private keys is one of the biggest factors resulting in the theft and misuse of code signing credentials. These keys might be stored on a developer's desktop computer, a build server, a virtual server or even a web server connected directly to the internet. These practices expose extremely critical assets to cybercriminals or even rogue employees within the company.

Without global visibility, it is hard for InfoSec teams to understand the magnitude of risk caused by unprotected private keys. Delegating the control of private key access to the developers drastically increases the chance of compromise and provides little recourse for mitigation and damage control.

Lack of Policy Enforcement

Even when companies establish policies and procedures for code signing operations, their InfoSec teams may have difficulty enforcing them. One challenge is that procedures often vary on a project-by-project basis to address these various elements:

- Type of software project
- Software tools used to sign code
- Types of code signing certificates used
- Who needs to approve the usage of code signing certificates

There are other challenges that impact policy enforcement:

- When approvals are required before code is signed, InfoSec teams need to ensure those approvals are obtained.
- Only signing certificates issued by a set of whitelisted CAs should be used.

To be effective, policy enforcement needs to be built into automated code signing processes and workflows. However, most organizations use platforms that do not support process automation, making enforcement problematic and haphazard.

InfoSec Bottleneck

For a few organizations, these problems don't apply because their InfoSec teams are already responsible for controlling all code signing operations. In these rare cases, these organizations often have a different problem—their InfoSec teams are a bottleneck for software releases. As software development moves towards shorter development lifecycles with DevOps, Agile or Continuous Delivery methods, the need to sign code in a matter of milliseconds becomes crucial. If an InfoSec team is supporting hundreds of developers and is manually managing all code signing requests, then the team is undoubtedly a bottleneck for development.

5 Steps to Secure Your Code Signing Process

InfoSec teams can take steps to secure the enterprise's code signing process while at the same time offering software development teams a convenient and enabling code signing service—one that allows developers to work at speeds as fast or faster than they have experienced with no controls in place.

1. Establish Global Visibility

The first step in securing the code signing process is for InfoSec teams to establish complete visibility across the enterprise for all code signing activities:

- Know what code is being signed, no matter if it is for internal or external use
- Know what code signing certificates are being used and the certificate authority (CA) they come from, including internally generated certificates
- Know who signed the code, what machine it was signed on, when it was signed and with what code signing tool
- Know who, if anyone, approved the code signing operation

2. Centrally Secure All Code Signing Private Keys

The next step is to move private code signing keys off all developers' computers, build servers, web servers, etc., into an encrypted, secure and centralized storage location. Private keys should never leave this location for any reason. Companies need to provide storage options for developers based on the type of code signing certificates they are using. For example, extended validation (EV) certificates must be stored in a hardware security module (HSM). In addition, some compliance regulations may require that the private keys reside within a certain location. Therefore, each organization may need to offer multiple secure locations to address all requirements.

3. Define and Enforce Policy with Automation

One of the reasons development teams resist corporate policies is that most involve manual processes and don't support their specific project's needs. Because of this, organizations should provide a code signing platform that allows every development team to define their code signing policies and workflows:

- Who is authorized to sign code
- What code signing tools are authorized
- What certificates are allowed
- Who must approve the code signing based on certificate type or phase of software development

This platform then needs to automatically enforce those defined policies through workflow automation. To help ensure adoption, the platform should integrate with third-party code signing tools already in use and corporate platforms, such as Active Directory, ticketing systems and other SIEM tools.

4. Enable Speed and Ease of Code Signing

It is an undesirable burden for software development teams to manage their code signing activities—figuring out which certificates should be used in certain situations, requesting a code signing certificate from a CA, renewals, etc. This takes time and expertise. If InfoSec teams are able to replace this burden with an automated, flexible platform that supports their needs, development teams will be able to achieve their goal of faster software delivery. But keep in mind, the solution that is offered must not introduce new burdens: requiring build scripts to be written, memory-resident programs to be installed, new tools or APIs to be learned, or slowing down development teams' automated build processes by requiring the code to be offloaded to a central server for signing.

5. Show Compliance

InfoSec teams are charged with securing the company's information and data, including code signing credentials. They must be able to show that they are effectively achieving this end goal via a secure code signing process across the entire enterprise. By having an irrefutable record of all code signing operations, knowing where all private keys are stored and knowing that policies are always enforced helps ensure and demonstrate compliance.

Venafi CodeSign Protect

Venafi CodeSign Protect offers capabilities that implement all five of the steps above for securing code signing processes. In addition, it is designed to scale from small teams to teams with tens of thousands of developers. Plus, the rich Venafi ecosystem of partner integrations ensures that the Venafi Platform works with the most popular CAs, HSMs, DevOps tools, software development tools and build infrastructure environments.

Venafi CodeSign Protect

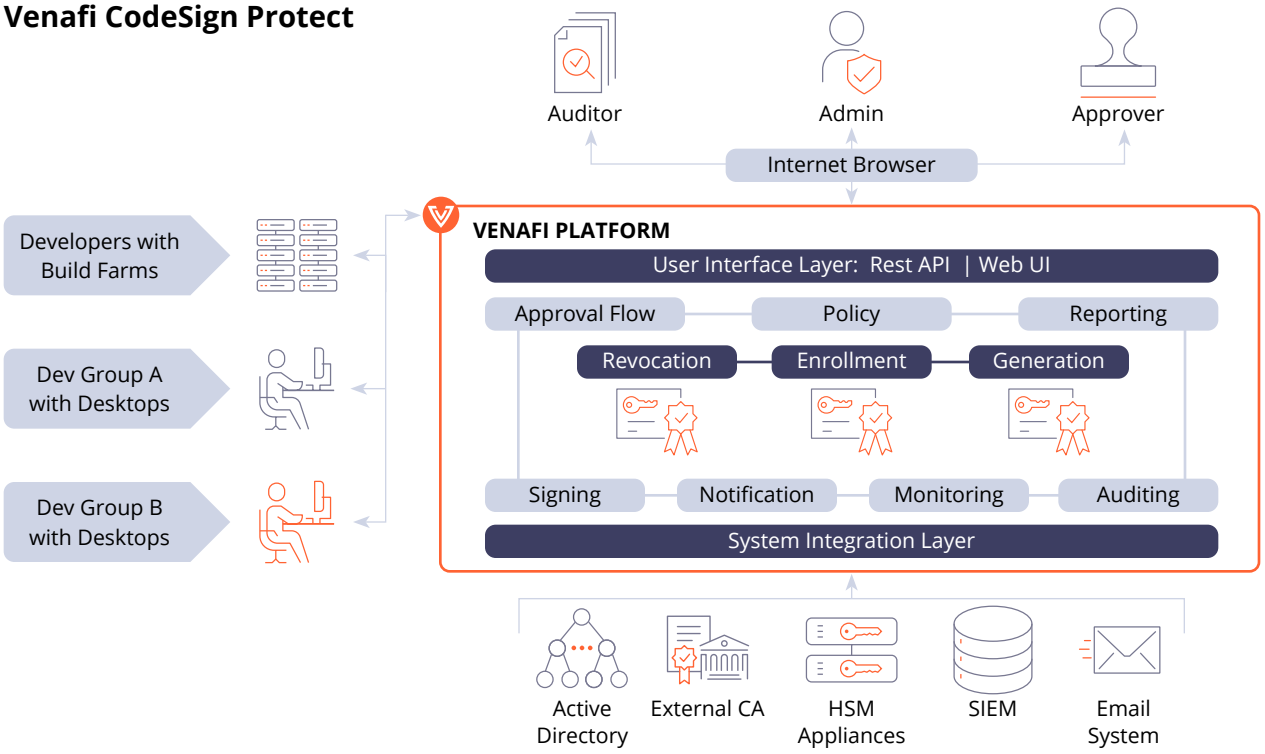


Figure 1: How CodeSign Protect Fits Within a Network Infrastructure

CodeSign Protect uses a role-based approach that allows software development teams to define their own code signing policies and workflows. It then automates those workflows and provides assurance to the InfoSec team that all policies are enforced and code signing private keys remain protected in a secure, centralized location.

Unify Machine Identity Management

Venafi provides protection across different types of machine identities, including keys and certificates that are used as code signing credentials. To protect code signing keys and certificates, Venafi has designed Venafi CodeSign Protect from the ground up to secure all aspects of the code signing process without adding additional burden on software development teams. CodeSign Protect is integrated with the Venafi Platform—the same platform that many of today's largest enterprises use to protect their companies' SSL/TLS, SSH and endpoint machine identities.

Learn how CodeSign Protect can give your InfoSec team the visibility and control needed to secure your code signing process while also supporting the goals and practices of your developers.

Visit **venafi.com/solutions/initiative/code-signing** for more resources and product information.

Trusted by

5 OF THE 5 Top U.S. Health Insurers
5 OF THE 5 Top U.S. Airlines
3 OF THE 5 Top U.S. Retailers
3 OF THE 5 Top Accounting/Consulting Firms
4 OF THE 5 Top Payment Card Issuers
4 OF THE 5 Top U.S. Banks
4 OF THE 5 Top U.K. Banks
4 OF THE 5 Top S. African Banks
4 OF THE 5 Top AU Banks

Venafi is the cybersecurity market leader in identity management for machines. From the ground to the cloud, Venafi solutions automate the lifecycle of identities for all types of machines—from physical devices to software applications, APIs and containers. With more than 30 patents, Venafi delivers innovative solutions for the most demanding, security-conscious organizations in the world. **To learn more, visit venafi.com**