KEYFACTOR

SECURE EVERY DIGITAL IDENTITY

EBOOK

The Definitive Roadmap to Secure Code Signing

PRACTICAL GUIDANCE FOR SOFTWARE DEVELOPERS AND IOT MANUFACTURERS

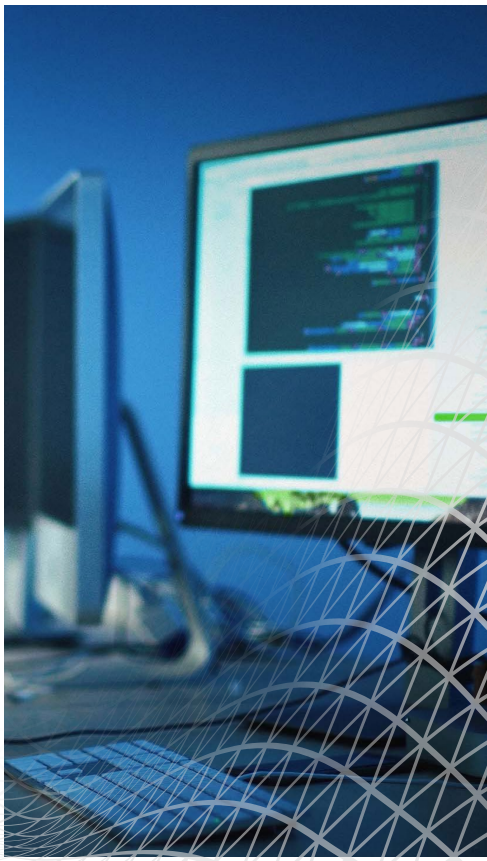

Table of Contents

The Power of Code Signing

Software and firmware developers must digitally sign code to establish trust and integrity in their products. With the rise of the Internet of Things (IoT) and explosive growth of software and mobile apps, the process of code signing is now more important than ever.

INTRODUCTION

We live in a world that runs on code. Software permeates virtually every aspect of our lives, from the things we use each day to the critical infrastructure of our society. It's difficult to imagine a business today that doesn't depend on software in some way. As the IoT continues to grow, software only becomes more embedded in our physical world.

There was a time when users could trust the software they downloaded, but that is far from true today. As hackers become increasingly adept in the art of spreading malware, even IT professionals find it difficult to know whether the software or the product they've purchased is legitimate. This is where code signing comes in.

Code signing is a cryptographic method used by developers to prove that a piece of software is authentic. By digitally signing apps, software, or embedded firmware with a private key, proof is provided to end users that the code originates from a trusted and legitimate source and that it hasn't been tampered with since it was published.

> "When you sign a piece of code, you make a statement that the software came from your organization and that you stand behind it."

– TED SHORTER, CTO & CO-FOUNDER, KEYFACTOR

HOW CODE SIGNING IS USED

ENTERPRISE IT

Enterprise IT teams must ensure that any internal scripts or utilities applied across the business are signed to prevent tampering by internal users or external threats.

MOBILE APP DEVELOPERS

Popular app stores from Microsoft, Google, and Apple require mobile apps to be signed before they can be submitted and published for purchase.

SOFTWARE VENDORS

Developers of software are often required to sign code to support installation. Operating systems like Windows and macOS will warn users if software or drivers are not signed.

IOT MANUFACTURERS

In the emerging Internet of Things (IoT), code signing is the most effective way to ensure the integrity of devices from activation through firmware and software updates.

Why Security Matters

Code signing without securing your private keys can expose you to more risk than no code signing at all. Attackers seek to compromise these keys to sign and distribute malicious code to your customers – masked as legitimate software or firmware.

KEY THEFT

If the private keys linked to your code signing certificates are compromised, it's game over. Stolen code signing keys are top prize for hackers – either sold or used to create signed malware that appears be to published by your developers.

SIGNING BREACH

Hackers don't need your keys to sign malware. If build servers or developer workstations with unhindered access to code signing systems are breached, an attacker can simply submit malware to be signed and distributed without detection.

INTERNAL MISUSE

Developers specialize in code, not security. Code signing keys and certificates can easily be misused or misplaced by developers, making it much easier for would-be attackers to undermine the integrity of your code signing operations.

THE IMPACT OF CODE COMPROMISE

Software publishers, device manufacturers, and in-house development teams have adopted code signing as a way to protect their intellectual property, their company brand, and their end users. But the trust and integrity of code signing hinges entirely on the security of your keys.

A single breach in this "chain of trust" can bring your entire business to a halt. Efforts to quickly revoke and re-issue certificates, notify customers, and push out a newly signed update are expensive – not to mention the immediate revenue loss and the cost of re-establishing trust with your users, partners, and investors.

> "Organizations experienced an average of 4 incidents involving the misuse of code signing keys and certificates in the past two years. The average cost for these incidents was $15 million."[1]

Sources:
1. https://info.keyfactor.com/the-impact-of-unsecured-digital-identities-ponemon-report

Know Your Risks

Recent code signing attacks underscore the importance of managing reputational risk. Whether you consume software or sell it, all business leaders need to invest in the trust associated with their digital brand – and expect the same of their vendors.

ADOBE BREACH

REPORTED: 2012
BREACH

Hackers breached a build server with access to the Adobe code signing system. Once inside, the hackers used the server to sign malware with a valid code signing certificate. Adobe responded by revoking the certificate, impacting three of their applications and millions of users.[2]

BIT9 INCIDENT

REPORTED: 2013
THEFT

Bit9 received reports from its customers that malware was discovered in their networks – malware that was digitally signed by Bit9's own code signing certificate. It was later discovered that hackers infiltrated a virtual machine and stole a certificate to sign and distribute the malware to at least three Bit9 customers.[3]

D-LINK LEAK

REPORTED: 2015
MISUSE

Hackers don't always have to steal a valid code signing certificate, especially when manufacturers unknowingly publish it themselves. In 2015, developers at network equipment manufacturer D-Link accidentally published four private code signing keys in open-source firmware. No known malware was signed with these keys, but D-Link received harsh criticism for the incident.[4]

Sources:
2. https://www.wired.com/2012/09/adobe-digital-cert-hacked/
3. https://krebsonsecurity.com/2013/02/security-firm-bit9-hacked-used-to-spread-malware/
4. https://www.engadget.com/2015/09/18/leaked-d-link-code-signing-key-could-make-malware-look-legit/?guccounter=1

Know Your Risks (Cont.)

> "We saw the updates come down from the Live Update ASUS server. They were trojanized, or malicious updates, and they were signed by ASUS."[8]

SUCKFLY APT
REPORTED: 2016
THEFT

An Advanced Persistent Threat (APT) group based in China, known as Suckfly, stole code signing certificates from at least nine different companies in South Korea. The group then used these certificates over two years to sign hacktools and malware in a targeted campaign against companies in India.[5]

D-LINK (AGAIN)
REPORTED: 2018
THEFT

Taiwan-based tech companies D-Link and Changing Information Technology were targeted by another APT group known as BlackTech. At least two code signing certificates were stolen and used to sign Windows malware known as PLEAD, which steals passwords entered into the web browsers of infected machines.[6]

SHADOWHAMMER
REPORTED: 2019
BREACH

ASUS – a well-known manufacturer of laptops and mobile phones – unknowingly pushed malware to thousands of its customers for at least five months in a sophisticated software supply chain attack dubbed "Operation ShadowHammer." Hackers compromised two ASUS code signing certificates and pushed out signed malware through the ASUS Live Update Utility, inserting backdoors into at least 1 million devices.[7]

Sources:
5. https://www.symantec.com/connect/blogs/suckfly-revealing-secret-life-your-code-signing-certificates
6. https://www.welivesecurity.com/2018/07/09/certificates-stolen-taiwanese-tech-companies-plead-malware-campaign/
7. https://blog.keyfactor.com/hackers-hijacked-asus-software-updates-to-install-the-need-for-code-signing
8. https://www.vice.com/en_us/article/pan9wn/hackers-hijacked-asus-software-updates-to-install-backdoors-on-thousands-of-computers?__hssc=1892590.6.1560255860314&__hstc=1892590.f79945add34c736eb171fecab8242cdf. 1543938429465.1560250778383.1560255860314.340&__hsfp=1885817573&hsCtaTracking=77837cbb-5839-4f3f-bbc0-340255243964%7C7428567c-0f2d-47c3-aabd-5661ba30f65d

Roadblocks to Success

The good news is that most independent software vendors (ISVs) and device manufacturers recognize the importance of code signing – the . The biggest challenge is how to implement it in a way that effectively meets the needs of both developers and IT security teams.

SPEED VS. SECURITY

Security and PKI teams would prefer to isolate and lock down private keys, but developers need quick access to sign code and push it to production. The biggest problem is how to implement safeguards to prevent misuse of keys and certificates without impeding the productivity of your developers.

DEVOPS & AGILE DEVELOPMENT

DevOps practices have taken the IT world by storm. Fast and frequent incremental software builds are the name of the game. Any changes to the Software Development Lifecycle (SDLC) can introduce more risk than they aim to prevent. Security must adapt to existing DevOps workflows and signing processes.

DISPERSED DEV TEAMS

Today's development teams collaborate across globally dispersed locations. When a remote team needs to sign software code, the simple easy solution is to purchase a signing certificate. Once purchased, certificates are often left within reach of hackers and out of the purview of your security team on disparate developer workstations or build servers.

SOPHISTICATED THREATS

Software supply chain attacks are becoming increasingly frequent and sophisticated. Hackers, cybercriminals, and even state-sponsored attacks put the security and integrity of your software at risk. As the cost of code signing certificates on the underground market continues to rise, some attackers have taken a more direct approach.

YOUR ROADMAP TO SECURE CODE SIGNING STARTS HERE

Enough theory. Let's dive into the four practical steps your organization can take to overcome these challenges and the right solution to help you get there.

01 Safeguard Private Keys

Private keys that developers use to sign code are invaluable to hackers. Once compromised, these keys can be used to sign virtually any code and distribute it to thousands of users. These types of attacks have become more and more frequent, as hackers seek to evade malware detection tools.

No industry or enterprise is immune – even trusted vendors in security and hardware such as Bit9 (now Carbon Black) and D-Link have fallen prey to persistent attackers that successfully weaved through their network to find and compromise code signing keys, and used them against their own customers.

The burden to sign code often falls on developers that specialize in writing code, not securing keys. As a result, keys wind up in unsecured network locations such as developer workstations, build servers, and who knows where else. IT security teams are often left unaware of exactly how many code signing certificates they have or where they are stored.

GET A COMPLETE INVENTORY OF YOUR LANDSCAPE.

Digital certificates and keys used for code signing are high-value assets, yet 71% of organizations don't know exactly how many they have. Start first by taking an inventory of how many code signing keys you have, where they live, and how they are stored.

CENTRALIZE MANAGEMENT OF KEYS & CERTIFICATES.

Local code signing creates siloes in security and increases costs. A centralized, server-side solution can simplify administration, improve security, and eliminate the need to have a separate code signing certificate for every developer or build stream.

STORE PRIVATE KEYS IN A SECURE, CERTIFIED HSM.

Hardware security modules (HSM) are the most effective way to ensure that your private keys remain under your control. Keys can be stored or generated inside the HSM and used to sign code anywhere without ever having to leave the hardware.

CHOOSE A FLEXIBLE SOLUTION.

Choose a solution that is easy to deploy and scale as demand for code signing grows. Cloud HSM services offer all the benefits of hardware-level security, but without the upfront expense, manual set-up, and ongoing maintenance of dedicated hardware HSMs.

02 Secure Signing Operations

If a hacker breaches your developer network, they don't need to steal your keys. By gaining access to a build server or developer workstation with access to code signing infrastructure, hackers can simply submit malware to be signed and distributed without detection.

Better known as "software supply chain attacks," these threats are even more difficult to detect, because they often involve either an insider or an attacker with direct access to code signing. Even the likes of Adobe and ASUS with sophisticated security teams were unable to detect breaches in their code signing infrastructure for months.

Attackers will always find the path of least resistance. Storing private keys in an HSM reduces the risk of key theft, but you can bet that determined threat actors will find another way. In the case of Adobe, private keys were stored securely in an HSM, but hackers instead gained access to a build server and simply requested signatures for malicious code.

It's absolutely critical to ensure that only the right developers can sign the right code – at the right time. By allowing only authorized users to sign and approve code, you can ensure that even if a hacker does breach your network, they can't gain access to your critical code signing infrastructure.

DEFINE ROLES & SEPARATE DUTIES.

Establish separate roles for those authorized to submit code for signing and those authorized to approve signing requests. Dividing these duties will ultimately help ensure that only trusted users can sign code, keeping hackers out of signing operations.

ENFORCE CODE SIGNING POLICIES & PRACTICES.

Ensure that even authorized developers are only granted access to sign code for a defined duration of time, number of signatures, and other parameters to prevent illegitimate signing or internal misuse. Look for solutions that can help you enforce these policies.

SEGMENT TEST & RELEASE SIGNING.

If code is signed during the development and testing stages, certificates should be distinct from those used in production signing. Ensure that development keys are not linked to the same root of trust as keys used to sign production code.

TRACK SIGNING ACTIVITIES.

Development or release managers should track every use of private keys to sign code. This will help you to ensure that only the right developers are signing the right code, with the right keys,

03 Integrate with DevOps

It's no knock against developers to say that security isn't their top priority. In order to stay ahead of the digital curve, development teams must move fast to write code and push it to production. However, security teams are putting more and more pressure on developers to meet requirements that often clog up the build and release process.

Responsibility to secure code signing keys shouldn't fall on developers – they simply need access to submit and sign code quickly. It's about finding the right balance between IT security requirements to lock away private keys and developer needs to sign any code, from anywhere, without disruption.

FORMALIZE YOUR CODE SIGNING PROCESS.

Document, track, and rigorously follow the steps required to sign code as part of your software development lifecycle (SDLC). Define what checks and verifications (i.e. QA, pen tests, virus scans, static analysis, etc.) must be performed before signing.

MINIMIZE CHANGES TO THE SDLC.

Take a collaborative approach to ensure that security and development teams achieve mutual goals to protect keys without disrupting the SDLC. It's important to find solutions that adapt to your processes, not the other way around.

COVER DISTRIBUTED DEV TEAMS.

The ability to enable remote signing is critical with todays distributed development teams. Choose a solution that enables developers to sign code from anywhere, without the keys ever leaving the secure confines of hardware protection.

> "Security and risk management leaders must adhere to the collaborative, agile nature of DevOps to be seamless and transparent in the development process."[9]

INTEGRATE WITH SIGNING TOOLS & PLATFORMS.

Directly integrate code signing processes with existing tools and workflows whenever possible. Code signing should support multiple code file formats, platforms (i.e. Microsoft Authenticode, Java, etc.), and certificate authorities (CA).

Sources:
9. https://www.gartner.com/en/conferences/la/infrastructure-operations-cloud-brazil/rn-devsecops

04 Monitor & Audit Compliance

Implementing security is one thing, but keeping developers in compliance with your code signing policies and best practices is another. Code signing security is a continuous process, not a set-and-forget deployment. Certificates inevitably expire, keys and algorithms weaken over time, and threats continue to evolve. Security teams must be able to identify these risks quickly and effectively respond.

MONITOR CODE SIGNING OPERATIONS.

By monitoring code signing requests, authorizations, and signatures in real-time, security teams can more effectively detect anomalous activity and respond within minutes, rather than months.

LOG & AUDIT KEY USAGE.

Keep a comprehensive audit log of who used code signing keys, when, and who authorized the action. Review logs regularly for suspicious activity and ensure that these logs cannot be tampered with.

INCLUDE CODE SIGNING CERTIFICATES IN CLM.

All digital certificates in your organization should be governed by certificate lifecycle management (CLM), including Extended Validation and Standard Code Signing Certificates – with provisions for how they are requested, issued, renewed or revoked.

ENSURE

Keep a comprehensive audit log of who used code signing keys, when, and who authorized the action. Review logs regularly for suspicious activity and ensure that these logs cannot be tampered with.

Conclusion

Keyfactor™ Code Assure

SECURE CODE SIGNING AT THE SPEED OF DEVOPS

Code signing is critical, but it isn't enough on its own. Organizations must protect their keys and certificates and implement robust code signing practices. The price for falling short is a bad experience for your users and an even worse experience for your business. As operating systems, mobile app stores, and CAs develop stricter rules around how to request and use code signing certificates – organizations must adapt or fall behind.

THE SOLUTION

Keyfactor Code Assure is the only platform that gives you complete visibility, control, and protection of codes signing operations, without disruption to existing build and release workflows. Code signing certificates and keys are stored centrally in a tamper-resistant certificate HSM. Once inside, the private keys never leave the HSM. Robust APIs enable developers anywhere with quick and controlled access to perform code signing, while security teams retain a full audit trail of code signing activities.

PROTECT YOUR KEYS

Store code signing keys and certificates in a centralized and secure hardware security module (HSM). Once inside, the keys will remain unusable until they are unlocked for use by a designated owner.

CONTROL ACCESS

Enable developers with quick and controlled access to certificates for signing. Restrictions can be enforced to unlock certificates for a time duration, number of signatures, who can sign, and more.

ANY TEAM, ANYWHERE

Allow developers to sign code from anywhere in the world with a unique technology that enables distributed teams to sign code remotely while keys and certificates never leave the secure confines of your HSM.

NO DISRUPTION

Secure code signing operations from end to end without making any changes to your existing SDLC or CI/CD pipeline.

END-TO-END VISIBILITY

Get a complete and actionable audit trail of who used code signing certificates, when, and who authorized the action – all from a single console.

DEPLOY ANYWHERE

Available on premise or in the cloud with the power of Thales Cloud HSM On Demand built right into the platform. No re-engineering, no hardware – no problem.

GET STARTED TODAY

Learn more about how Keyfactor can help you get started on your roadmap to secure code signing today.

Talk to our experts ▸

ABOUT

KEYFACTOR

Keyfactor, formerly Certified Security Solutions (CSS), is a leading provider of secure digital identity management solutions that enables organizations to confirm authenticity, and ensure the right things are interacting in the right ways in our connected world.

CONTACT US

▸ www.keyfactor.com
▸ 216.785.2990