Code Signing
What You Don’t Secure Can Hurt You

RYAN SANDERS
PRODUCT LEAD

JACK PALIVODA
SOLUTIONS ENGINEER
From Enterprise IT to the IoT

**ENTREPRISE IT TEAMS**
Enterprise IT teams sign internally distributed executables, scripts, and critical software infrastructure.

**MOBILE APP DEVELOPERS**
Popular app stores including Microsoft, Google, and Apple require mobile apps to be signed before they can be submitted.

**SOFTWARE DEVELOPERS**
Software developers must sign code to support Windows and macOS, as well as continuous patches and updates.

**IoT MANUFACTURERS**
Code signing is the most effective way to ensure integrity of devices from secure boot to continuous firmware updates.
What it is, why it matters

PROTECTS YOUR CUSTOMERS
Cryptographically-based method to verify **authenticity** and **integrity** of your products.

PROTECTS YOUR BUSINESS
Code is critical to your business. Stop hackers from publishing code in your company name.

RELIRES ON SECURITY OF KEYS
Signing code without protecting private keys can create more risk than no signing at all.
It is not an exaggeration to consider private code signing keys as the keys to the business’s kingdom.

SANS INSTITUTE
The risks of insecure code signing

**KEY COMPROMISE**

Attackers are now skilled in the art of stealing private keys to sell on the dark web or sign malware.

**SIGNING BREACH**

But they don’t need keys to sign malware, they can simply breach your code signing process.

**INTERNAL MISUSE**

Developers specialize in code, not security. Keys can easily be found in the wrong place or the wrong hands.
Known attacks and mishaps

2010
- STUXNET

2011
- MALAYSIAN GOV’T

2012
- BIT9

2013
- ADOBE

2014
- D-LINK

2015
- SONY

2016
- SYNful KNOCK

2017
- SUCKFLY APT

2018
- D-LINK (AGAIN)

2019
- ASUS
- APT27
- APT41
The impact of insecure code signing

**COST TO REMEDIATE**
IT, security, and other staff resources must respond to the attack – find and replace keys.

**LOSS OF TRUST & REPUTATION**
If an attacker signs malware under your certificate, it erodes trust in your brand.

**LOSS OF REVENUE**
Due to direct downtime for customers, the business, and intellectual property fraud.

"Organizations experienced an average of 4 incidents involving the misuse of code signing keys and certificates over the past 24 months. The average cost is estimated at $15 million."

Keyfactor-Ponemon Institute Report, 2019
Industry Use Case
MULTINATIONAL TECH COMPANY

- Development teams in US East, West, and Israel
- Multiple build server solutions – TFS, Jenkins, etc.
- Multiple dev languages – .NET, C++, Java, iOS
- More than 100+ different products to be signed
- Certs deployed to build servers, managed manually
- Signing process manual and “effort greedy”
Ad hoc and “DIY” code signing

SECURITY GAPS
Private code signing keys are left vulnerable in build servers and on endpoint devices.

HIGHLY COMPLEX
Manual signing and approval steps cause delays and create unnecessary complexity.

NO CONTROL
Security and audit teams lack centralized control of policies or visibility of code signing certificates.

NOT SCALABLE
Smartcards/HSMs add security, but are expensive at scale, don’t prevent breaches in the signing process.
**CHALLENGES**

- Private Keys Vulnerable
- Limited Visibility
- Lack of Control
- Security is a bottleneck
- Manual Signing Processes
- Distributed Teams

**PRIORITIES**

- Prevent Misuse or Theft
- Centralize Control/Security
- Demonstrate Compliance
- Push Quality Code to Production
- Don’t Mess with My Tools
- Don’t Slow Me Down
Code Signing Best Practices
01 | Protect Your Keys

- **Locate all of your keys and certificates**
  71% of organizations don’t know exactly how many keys and certificates they have. Start first with discovery.

- **Centralize management and control**
  Local code signing creates siloes and increases costs. Use a server-side solution to simplify and improve security.

- **Store private keys in a certified HSM**
  Protect and store your keys in a HSM where developers can sign code anywhere without needing access to keys.

**ASK YOURSELF**

- Do you know how many code signing certificates you have?
- How do you protect your code signing keys?
- If you found malware signed by your company’s code signing certificate, where would you start looking for the source?
02 | Secure Signing Processes

- **Control access and separate duties**
  Establish separate roles for those authorized to submit code for signing and those authorized to approve.

- **Define code signing workflows**
  Not every software project is the same. Define approval workflows and define access to private signing keys.

- **Enforce code signing policies and practices**
  Ensure that only the right developers are signing the right code, with the right keys – and at the right time.

**ASK YOURSELF**

- Do you know who has access to your code signing certificates?
- Does your organization enforce code signing policy to define where keys are stored, who has access, and who can approve usage?
- How do you control access for third-parties or remote teams?
03 | Integrate with Dev

- Minimize changes to the SDLC
  Ensure that code signing can be effectively integrated into the SDLC in a seamless and transparent way.

- Cover distributed dev teams
  The ability to enable remote signing without losing control of keys is critical in distributed development environments.

- Integrate with signing tools & platforms
  Avoid forcing developers to leave their native workflow in order to securely sign code. Adopt a CSP/KSP model.

ASK YOURSELF

- Does code signing in your organization involve slow manual processes?
- Do your developers try to circumvent these processes? Or do you suspect that they may be?
Log and audit key usage
Keep a comprehensive log of who used code signing keys, when, and who authorized the action.

Don’t just audit, continuously monitor
Actively monitor code signing requests, authorizations, and signatures to detect anomalies.

Include code signing certificates in CLM
All digital certificates in your organization should be governed by a CLM strategy, including code signing.

ASK YOURSELF
- Can you detect if a certificate is misused or compromised?
- If an IT auditor or compliance officer checks your code signing processes, are you prepared to respond?
- How do you ensure that certificates are secure and up to date?
Resources
WHERE TO GET STARTED

NIST CSRC White Paper
Security Considerations for Code Signing

SANS Institute White Paper
The Scary and Terrible Code Signing Problem You Don’t Know You Have

Keyfactor White Paper
The Definitive Roadmap to Secure Code Signing

Scan Me
Questions?

LIVE DEMOS AT BOOTH #601
Thank You

SEE OUR TEAM AT BOOTH #601

ROB BERNHARDT
SALES DIRECTOR

JASON FIOROTTO
VP MARKETING

RYAN SANDERS
PRODUCT LEAD

JACK PALIVODA
SOLUTIONS ENGINEER