



Software Code Signing and Supply Chain Management

Gary DiPalma

CTO

Zeva Inc

Agenda

- Introduction: Importance of Code Signing in Supply Chains
- Supply Chain Threats & Attack Vectors
- Dive into Code Signing
- Securing the Supply Chain
- Governance & Risk Management
- Closing Thoughts & Q&A



Why Code Signing Matters



Ensures authenticity and integrity of software



Prevents unauthorized code changes



Establishes trust in the software supply chain



Supply Chain Attack Vectors



Source code injection



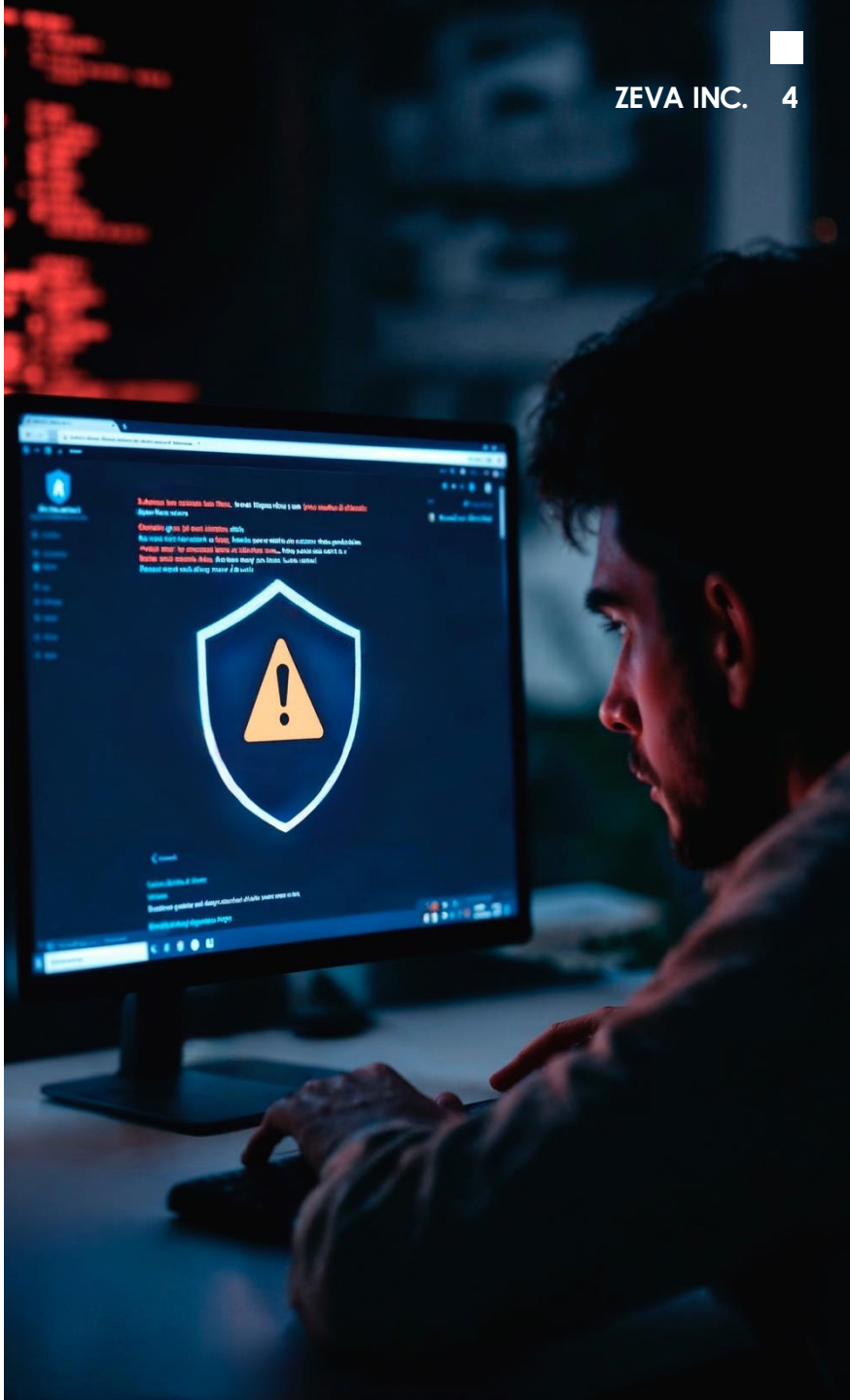
Build system compromise



Stolen certificates and key misuse



Malicious dependencies and update hijacking



Real World Examples

- **Malicious Code / Signing Breach**
 - SolarWinds Orion (2020)
 - 3CX (2023)
 - CCleaner (2017):
- **Key Theft**
 - NVIDIA signing cert leaked (2022)
 - GitHub Desktop & Atom certificates stolen (2022)
 - ASUS Live Update (2019)
- **Internal Misuse**
 - GitHub CodeQL Secret Leak (2025)
 - Mercedes-Benz GitHub token leak (2023)
 - D-Link Leak (2015)



Modern Code Signing Workflows



Hashing and private key encryption



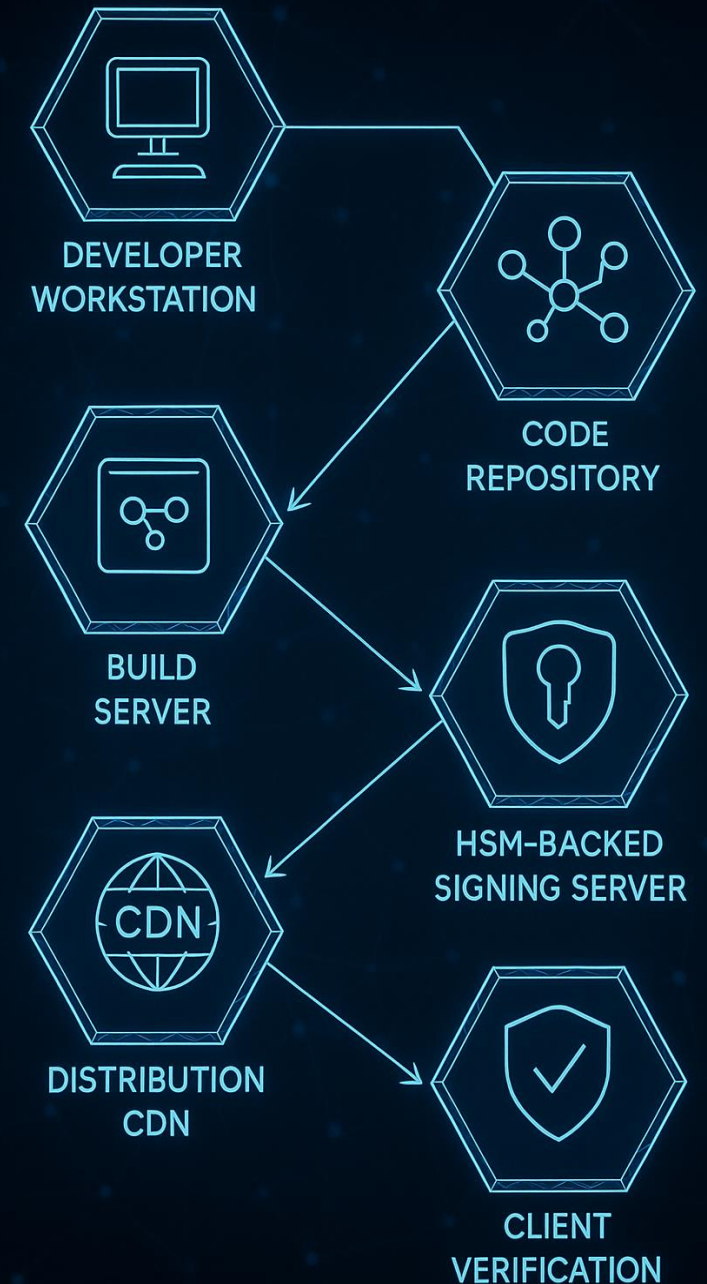
Certificate types: OV, EV, Self-signed



Timestamping to maintain validity



Integrated into CI/CD pipelines



Securing Code Signing Operations



Use HSMs or cloud KMS for key storage



Role-based access to signing systems



Automated and logged signing processes



Verify integrity before and after signing



Tools and Frameworks



CodeSigning Platform: Commit Level Signing, Keyless and Transparency



SLSA: Supply chain security maturity model



TUF: Resilient update framework



Integration with CI/CD and package managers



Binary and Run-Time Verification

Governance & Risk Management



Policy enforcement and training



Auditing and logging of all signing events



Certificate lifecycle and revocation management



Vendor security requirements and SBOMs



Closing Thoughts

- Code signing is essential for trust and integrity
- Must be combined with secure infrastructure
- Stay proactive with tools and community standards
- Enable collaboration across technical and policy teams



References

NIST Secure Software Development Framework (SSDF) – SP 800-218

<https://csrc.nist.gov/publications/detail/sp/800-218/final>

NIST SP 800-161r1 – Supply Chain Risk Management Practices

<https://csrc.nist.gov/publications/detail/sp/800-161/rev-1/final>

Executive Order 14028 – Improving the Nation’s Cybersecurity

<https://www.whitehouse.gov/briefing-room/statements-releases/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/>

ISO/IEC 27034 – Application Security Guidelines

<https://www.iso.org/standard/44378.html>

SLSA (Supply-chain Levels for Software Artifacts)

<https://slsa.dev>

The Update Framework (TUF)

<https://theupdateframework.io/>

References

SolarWinds SUNBURST Attack

<https://www.fireeye.com/blog/threat-research/2020/12/evasive-attacker-leverages-solarwinds-supply-chain-compromises.html>

<https://www.cisa.gov/news-events/alerts/2020/12/13/advanced-persistent-threat-compromise-government-agencies>

3CX Desktop App Attack (2023)

<https://www.mandiant.com/resources/blog/3cx-software-supply-chain-compromise>

<https://www.crowdstrike.com/blog/3cx-supply-chain-attack/>

ASUS Live Update (ShadowHammer)

<https://securelist.com/operation-shadowhammer/89992/>

CCleaner Breach (2017)

<https://blog.talosintelligence.com/ccleaner-backdoor/>

Stuxnet Case

https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf