IMPLEMENTING NIST SSDF WITHOUT KILLING YOUR CI/CD PRODUCTIVITY

NIST Secure Software Development Framework (SSDF v 1.1)

digicert®

ABOUT TODAY'S SPEAKER

Developed safety-critical avionics and medical software for 10+ years

Expert in software development methodology, cybersecurity, PKI

BSCS/EE, MBA

EDDIE GLENN

Senior Manager Software Trust, DigiCert

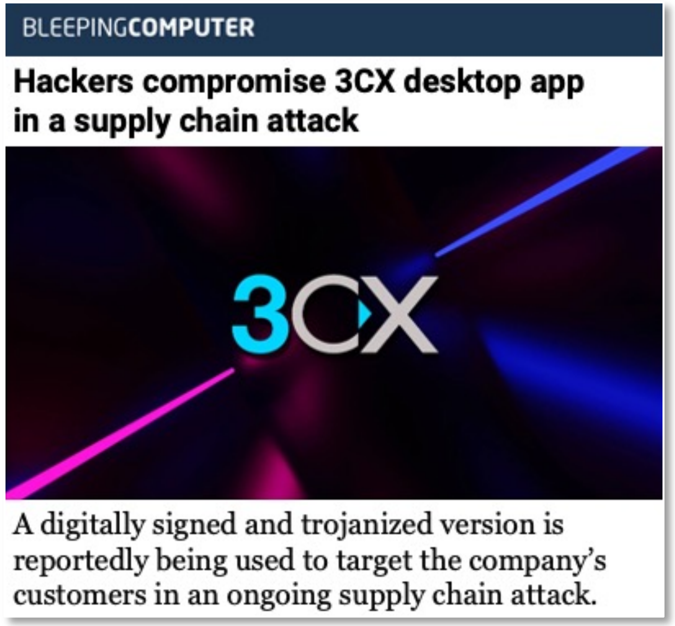
AGENDA

SOFTWARE ATE THE WORLD

PRACTICALLY EVERY BUSINESS IS A SOFTWARE BUSINESS.

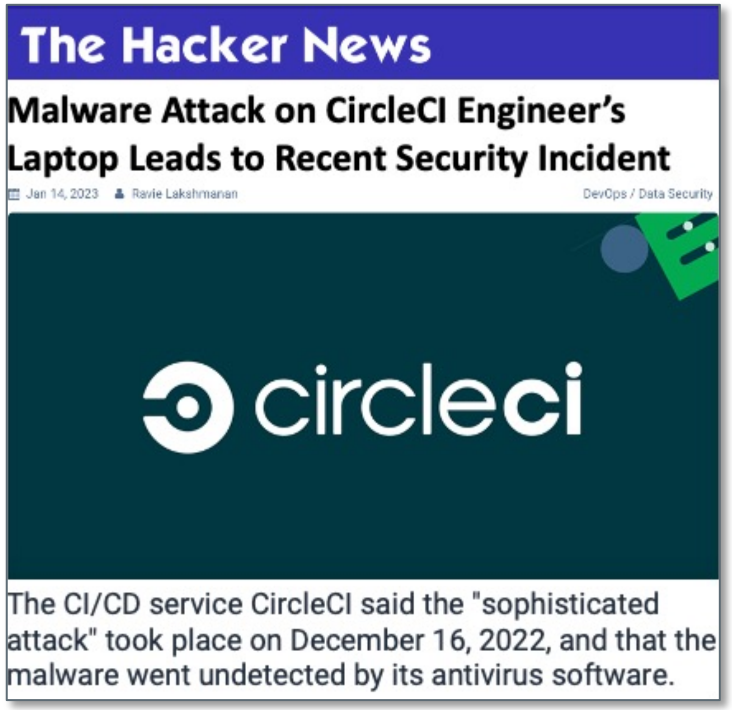
Financial | Healthcare | Transportation | Infrastructure | Retail | Agriculture | Industrial | Insurance | Communications | Tech | Entertainment

OUR SOFTWARE IS UNDER ATTACK


BLEEPINGCOMPUTER

Hackers compromise 3CX desktop app in a supply chain attack

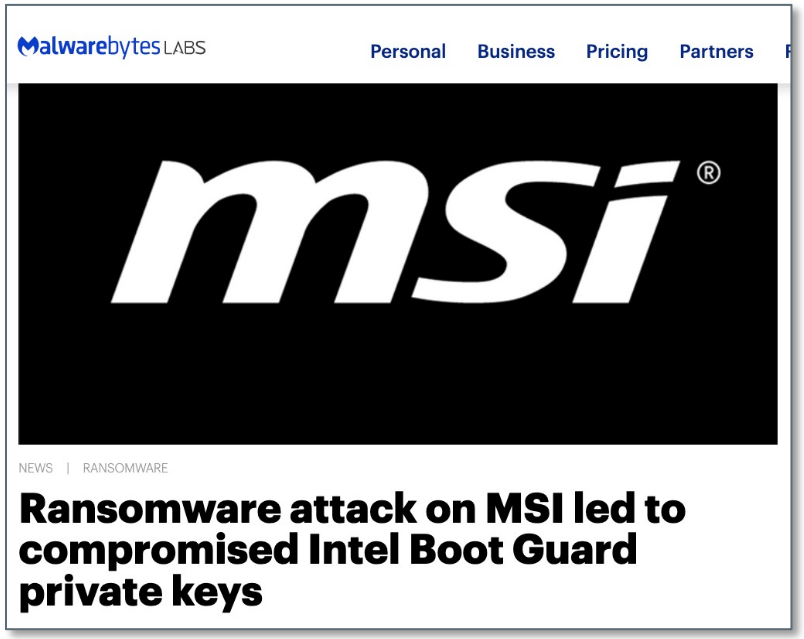
A digitally signed and trojanized version is reportedly being used to target the company's customers in an ongoing supply chain attack.


The Hacker News

Malware Attack on CircleCI Engineer's Laptop Leads to Recent Security Incident

Jan 14, 2023 Ravie Lakshmanan DevOps / Data Security

The CI/CD service CircleCI said the "sophisticated attack" took place on December 16, 2022, and that the malware went undetected by its antivirus software.


Malwarebytes LABS Personal Business Pricing Partners

NEWS | RANSOMWARE

Ransomware attack on MSI led to compromised Intel Boot Guard private keys

slack circleci okta Travis CI twilio HEROKU

91%

of businesses reported a software supply chain attack last year

-- Data Theorem

LOST REVENUE, MARKET SHARE, REPUTATION

NotPetya malware estimated by US Dept of Homeland Security to cause $10B in world-wide damages

LIABILTY

CIVIL FINES

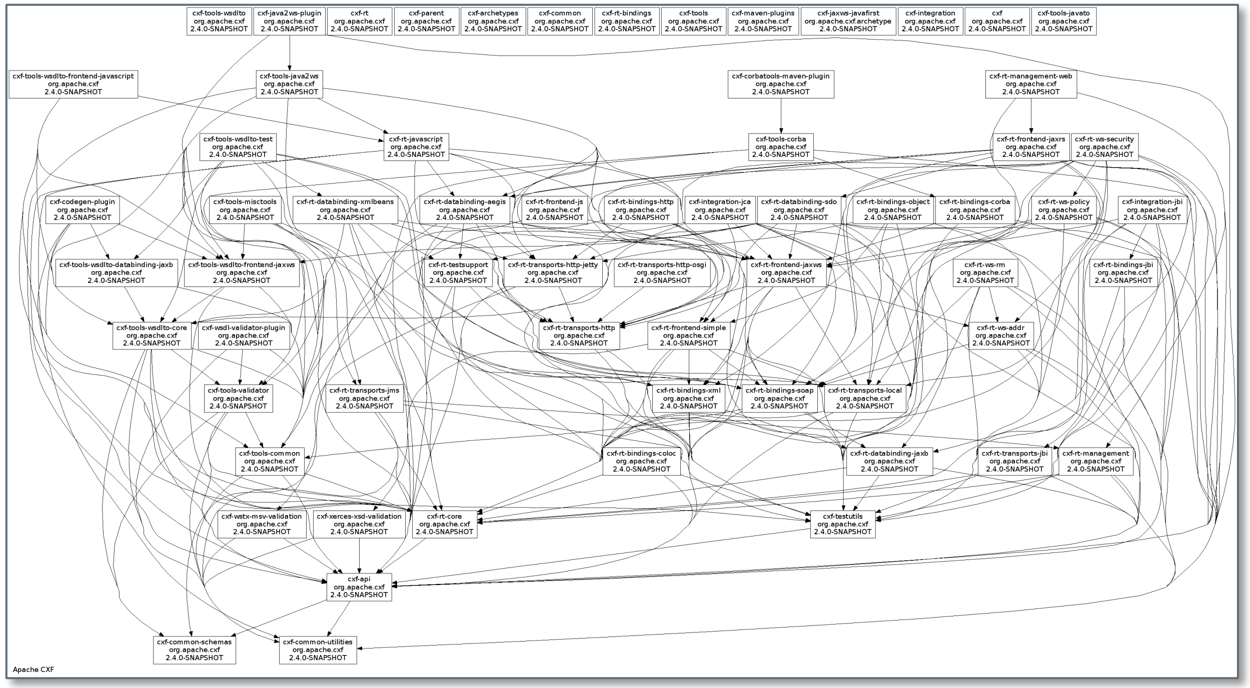
WHY IS STOPPING THESE ATTACKS SO HARD?

- Modern Software is Complex
- Organizations are Siloed & Understaffed
- Broad Attack Surfaces & Diverse Attacks

MODERN SOFTWARE IS COMPLEX

It's not your typical Windows app anymore…

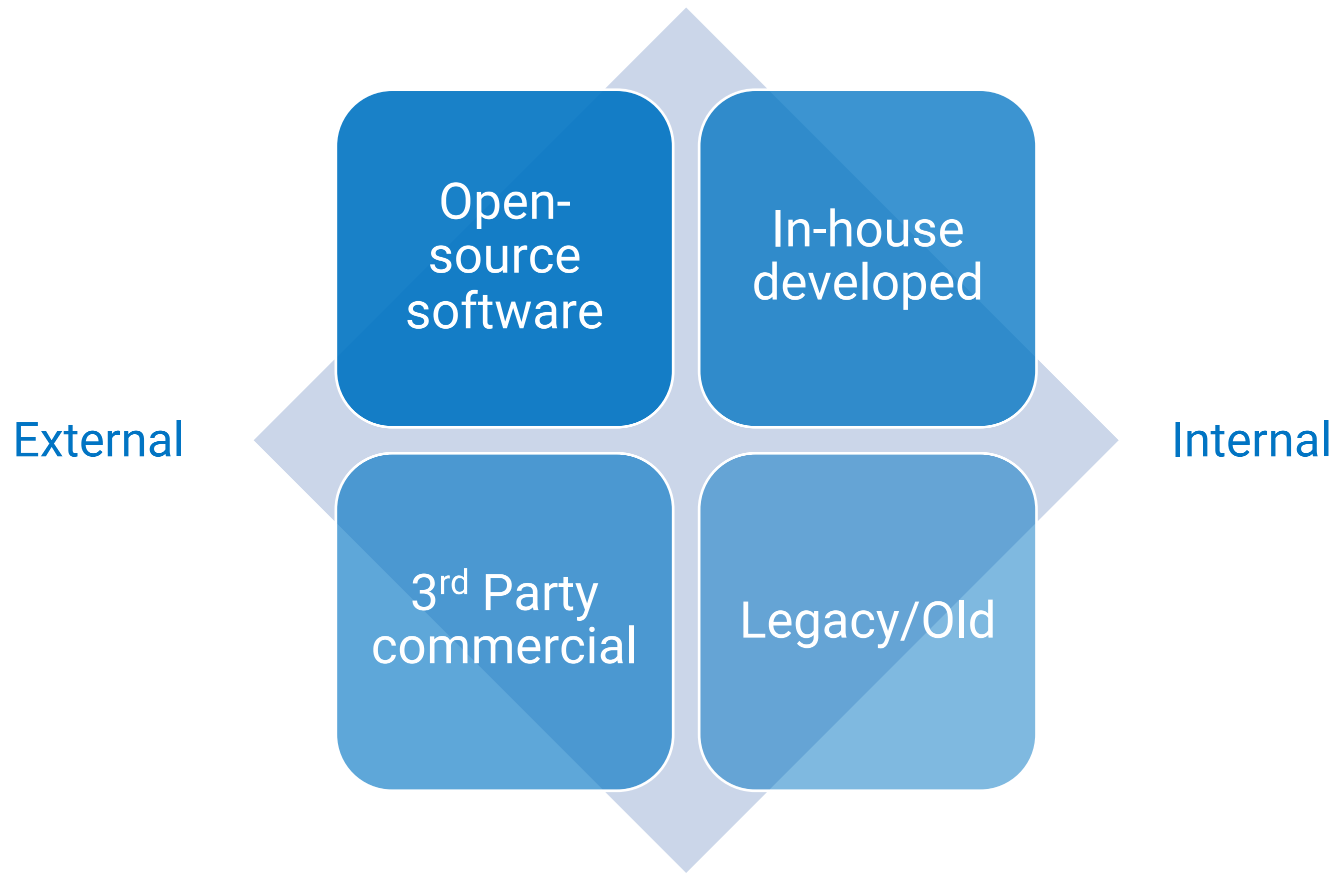
- Geographically distributed software teams
    - 100's or 1000's of global developers
- Extremely large code bases
- Multi-platform deployment
- Large software supply chain:
    - Open-source software
    - Third party commercial software
    - In-house developed software
    - OLD/Legacy Software
- Lots of dependencies, known & unknown
- DevOps/CI/CD – frequent releases
- Global presence – global regulatory issues



Apache HTTP Server Dependency Graph – approx. 2M SLOC

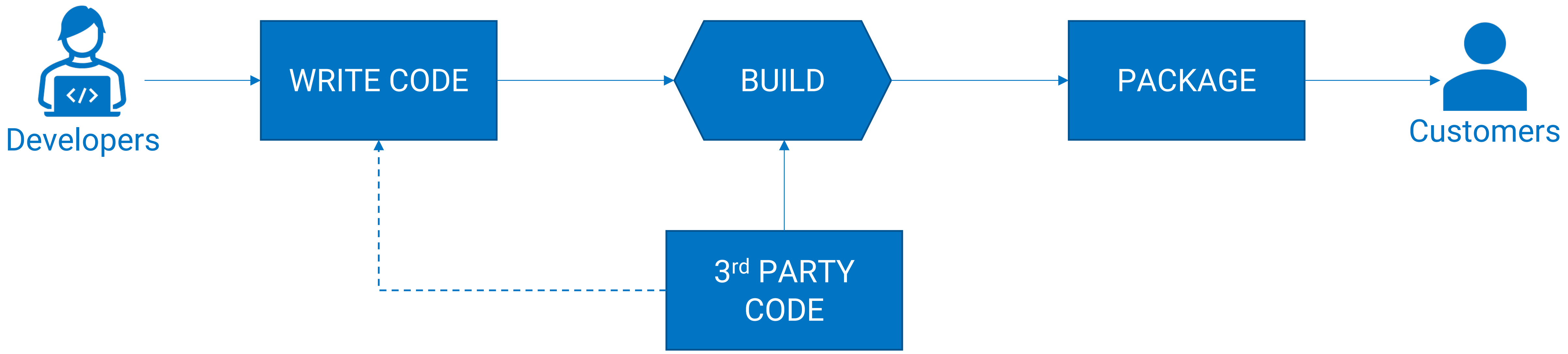
SOFTWARE SUPPLY CHAIN

Where does your software come from?



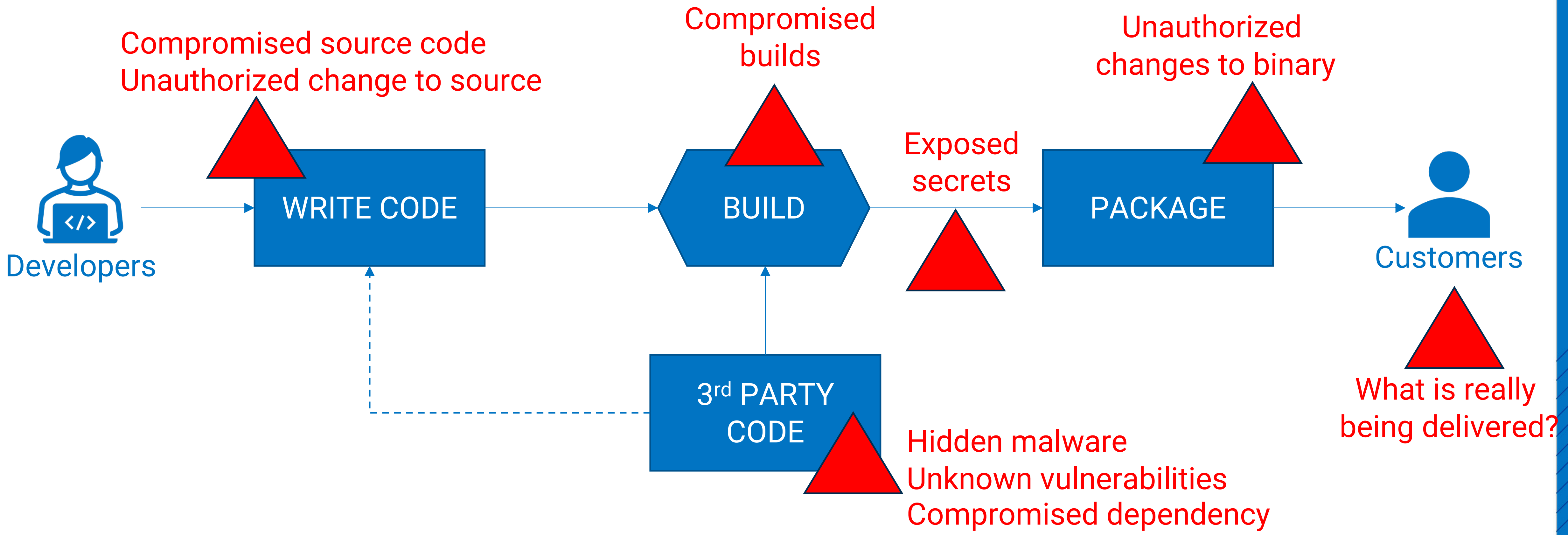
HOW IS SOFTWARE MADE?

Software Development Lifecycle



BROAD ATTACK SURFACE

Attacks can, and do, happen anywhere during this process



ORGANIZATIONAL CHALLENGES

Mobile App Dev Team | Linux Dev Team | Java Dev Team | Cloud App Dev Team | Windows Dev Team

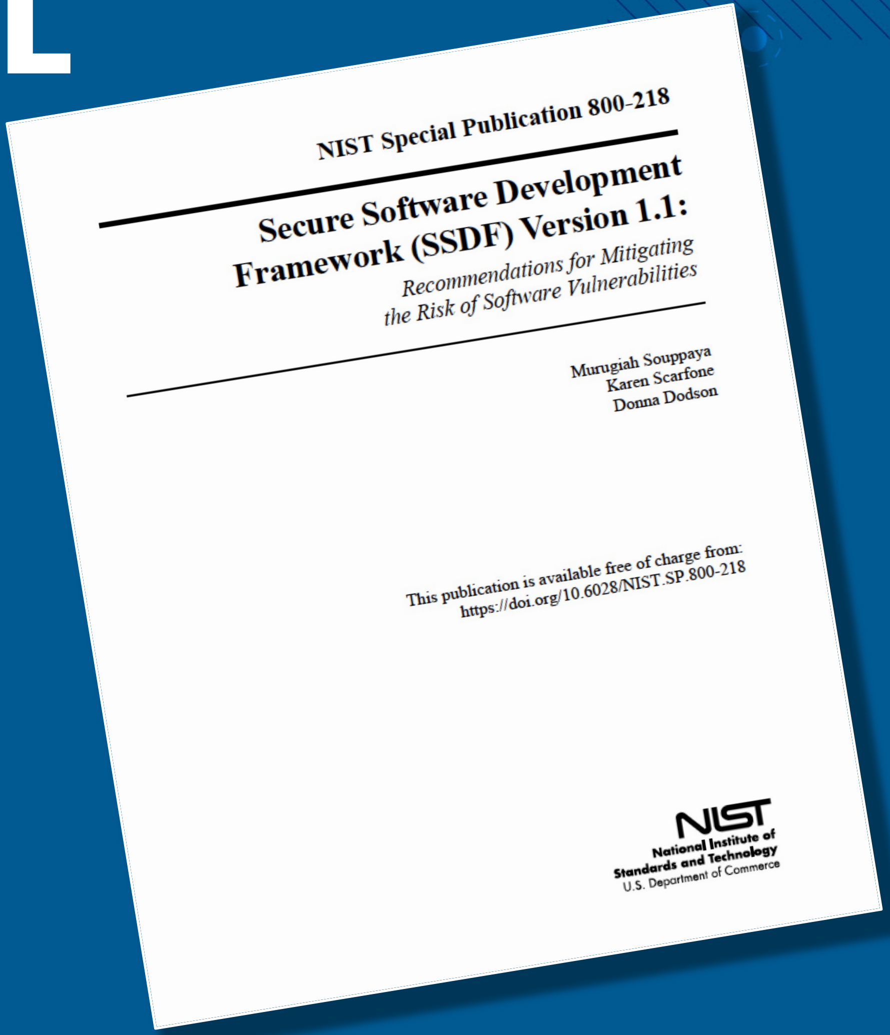
No Visibility & Enforcement | Successful Tampering | Missed Threats | Lack of Transparency

PKI Support | Product and Enterprise Security | Auditors, Risk, & Compliance

- Software complexity
  - Millions of lines of code
  - Open-source, 3rd party software
  - Legacy, old software
  - Multi-platform dev & deployment
- People
  - Disparate software teams, tools & methodologies
  - Pressure to do more in less time
  - Security often lower priority than new features
- Organization
  - Siloed teams – people, process, and technology
  - Product security & PKI support often understaffed
  - Security tools not integrated with dev tools

PRACTICAL GUIDANCE

for implementing
Secure Software Development Framework
(SSDF) V1.1


NIST Special Publication 800-218

Secure Software Development Framework (SSDF) Version 1.1:

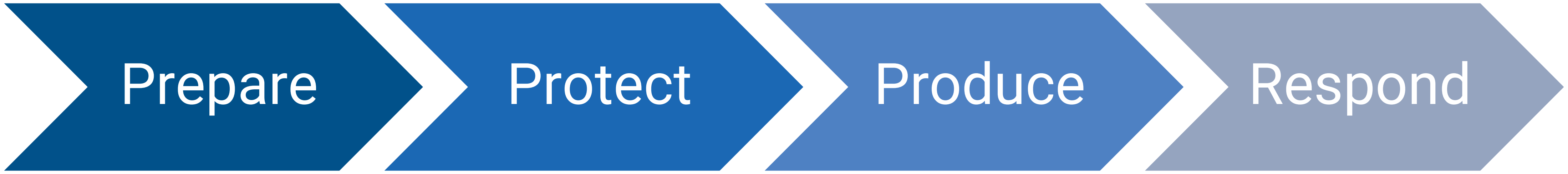
Recommendations for Mitigating the Risk of Software Vulnerabilities

Murugiah Souppaya
Karen Scarfone
Donna Dodson

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.800-218

NIST
National Institute of Standards and Technology
U.S. Department of Commerce

SSDF V1.1 AT A GLANCE



PREPARE THE ORGANIZATION (PO)

SDLC security should not be an afterthought

PO.1: Define security policies BEFORE software development begins including those that cover

- The software infrastructure used
- Software developed by the organization & developed by third parties (e.g., open source)

PO.2: Define security roles and responsibilities

- Across all aspects of the SDLC (requirements, design, testing, security)
- Define who is authorized to do what (e.g., sign code, approve code signing action)

PO.3: Select and utilize supporting SDLC tools

- Select to mitigate risks, automation
- Establish security policies on the tools
- Generate intermediate artifacts to support security policy

PLAN

PREPARE THE ORGANIZATION (PO)

SDLC security should not be an afterthought

PO.4: Define & use criteria for software security

- Ensure that criteria helps to manage risks (KPIs)
- Record security check approvals, rejections, and exception requests
- Use toolchains to automate tasks and gather information
- Automate decision making processes
- Only only authorized personnel to access information

PO.5: Implement and maintain secure SDLC environments

- Use multi-factor, risk-based authentication
- Network segmentation
- Minimize human access to toolchain systems
- Separate production from non-production systems

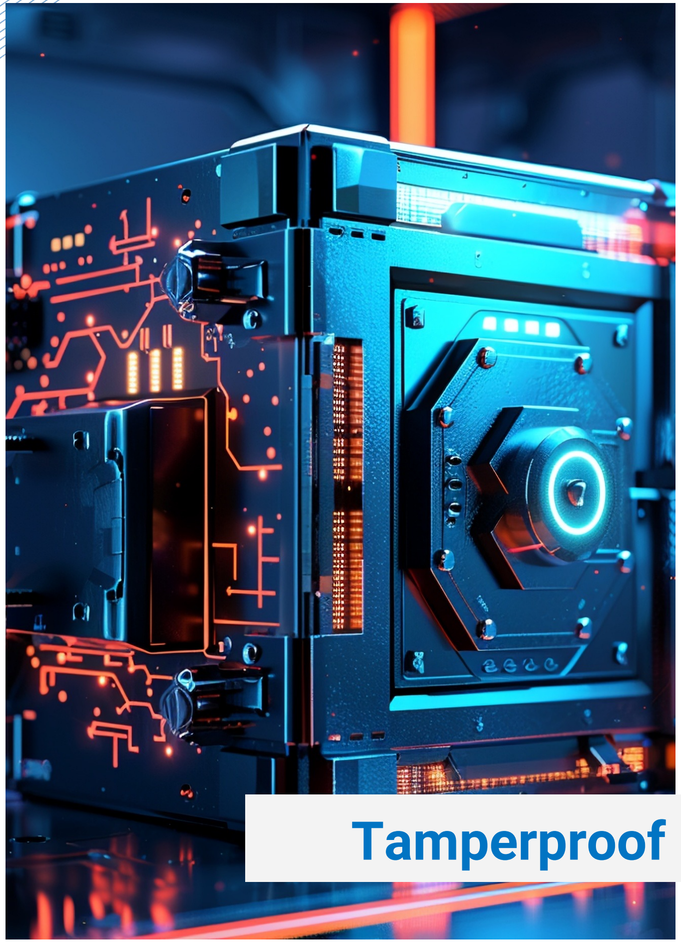
PLAN

PROTECT SOFTWARE (PS)

PS.1 Protect all SDLC artifacts from unauthorized access and tampering.

Store all SDLC artifacts (source, executables, scripts, CaC, etc) in a repository based on least privilege

- Store all artifacts in a repository and restrict access
- Digitally sign all SDFC artifacts – prevents unauthorized tampering and shows authenticity
- Use version control
- Have owners review and approve changes
- Use code signing to protect the integrity of executables
- Use cryptography to protect file integrity

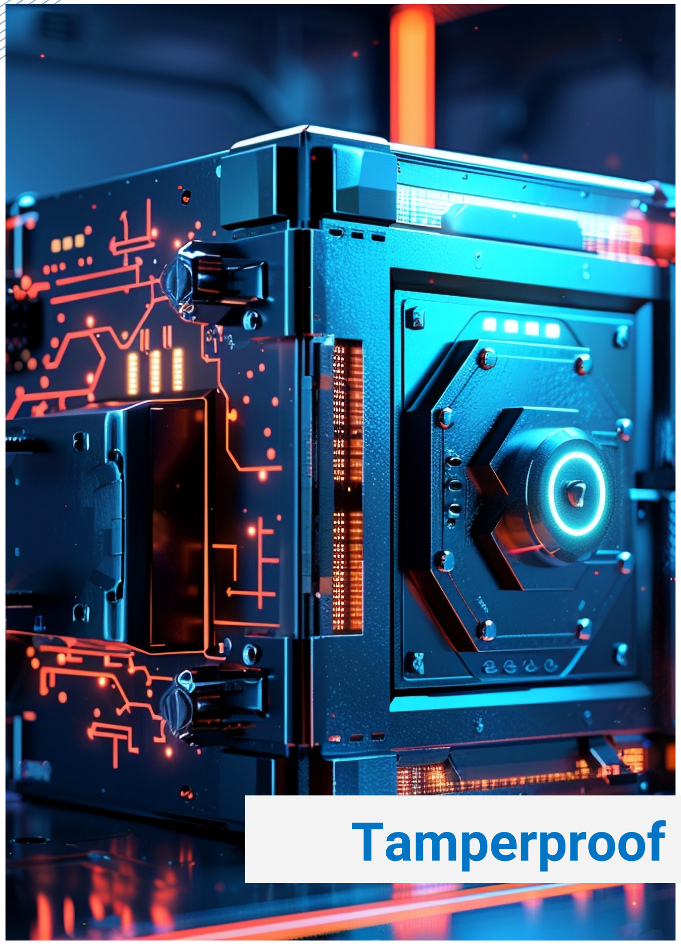


PROTECT SOFTWARE (PS)

PS.2 Provide a mechanism for verifying software release integrity

Make software integrity verification information available to software acquirers/consumers

- Post cryptographic hashes for release files
- Use an established certificate authority for code signing to consumers can trust can confirm the validity of your signatures
- Periodically review code signing processes, including certificate renewal, rotation, revocation, and protection (AUTOMATE!)



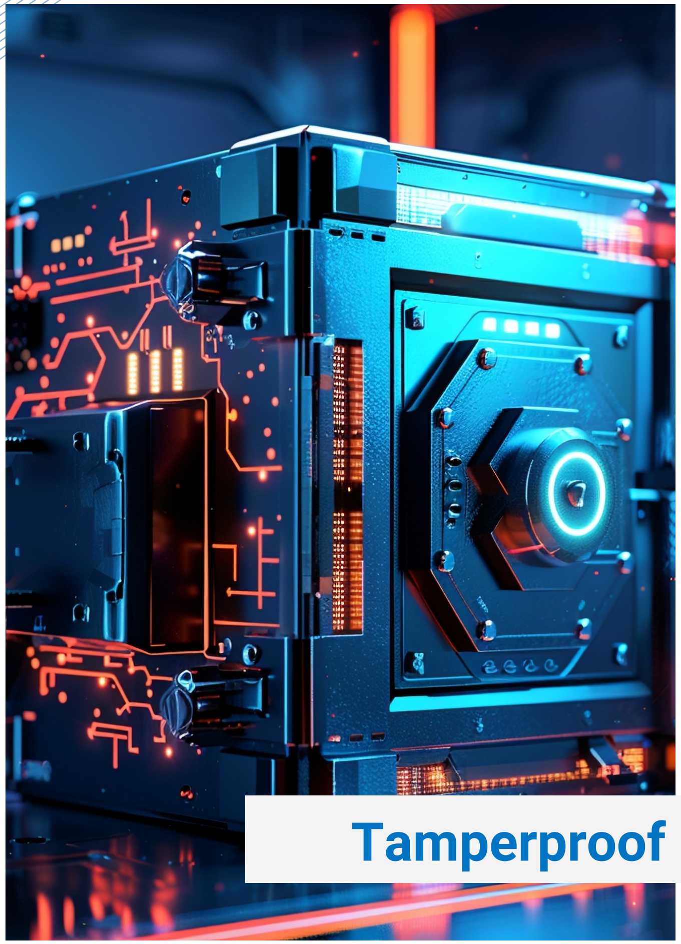
Tamperproof

PROTECT SOFTWARE (PS)

PS.3 Archive and protect each software release

Make software integrity verification information available to software acquirers/consumers

- Securely archive all files and supporting data for every software release, including integrity information
- Collect, share, and maintain provenance data for all components of each software release (SBOMs)



PRODUCE WELL-SECURED SOFTWARE (PW)

Design and write software to meet security requirements

- Utilize threat modeling/detection, attack modeling/detection
- Utilize built-in support for standardized security features instead of implementing proprietary ones
- Review software design for compliance with security requirements
- Verify that third-party software complies with security requirements
- Follow secure coding practices
- Use SDLC tools that improve executable security
- Perform code reviews to ensure that code adheres to security policies
- Test for security vulnerabilities

SECURE SOFTWARE

RESPOND TO VULNERABILITIES (RV)

Gather information from various sources about potential vulnerabilities that exist in software components used by your software

- Creating and maintaining SBOMs for each release imperative for this
- Monitor vulnerability databases
- Use tools to automate
- Create a policy that addresses vulnerability disclosures
- Plan and implement risk responses for vulnerabilities

RESPONSE PLAN

AUTOMATION IS KEY

when implementing frameworks like SSDF

NIST Special Publication 800-218

Secure Software Development Framework (SSDF) Version 1.1:

Recommendations for Mitigating the Risk of Software Vulnerabilities

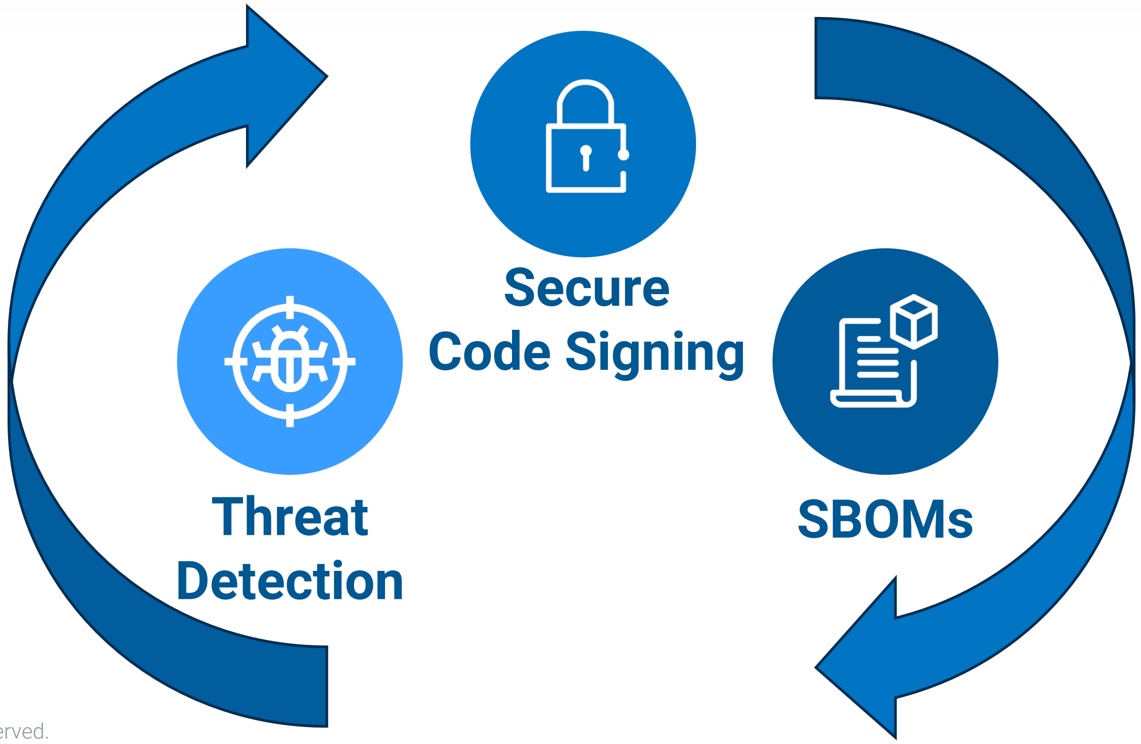
Murugiah Souppaya
Karen Scarfone
Donna Dodson

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.800-218

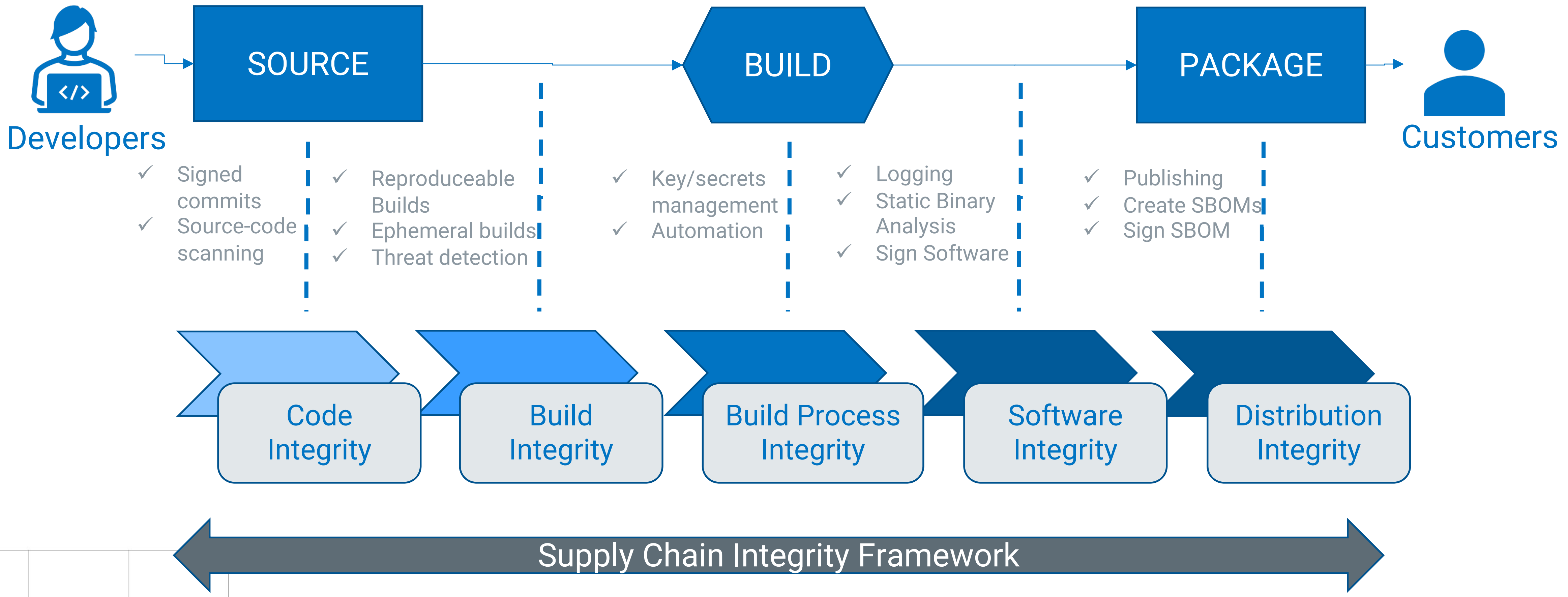
NIST
National Institute of Standards and Technology
U.S. Department of Commerce

EMBED THESE ACTIONS IN EVERY RELEASE CYCLE

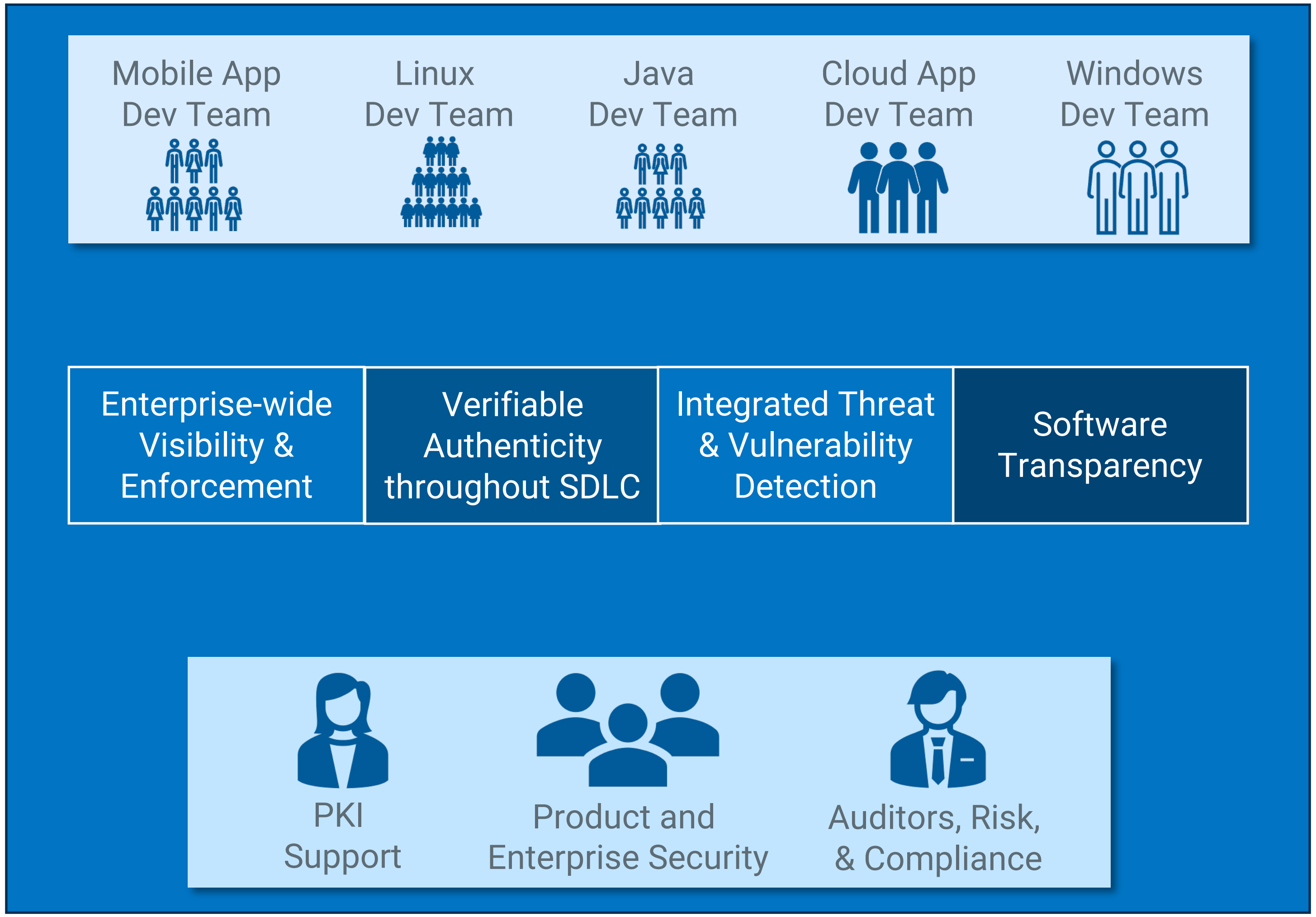
Threat detection, artifact signing & SBOMs in a unified security workflow



SOFTWARE INTEGRITY IN PRACTICE



DIGICERT'S APPROACH


Mobile App Dev Team
Linux Dev Team
Java Dev Team
Cloud App Dev Team
Windows Dev Team
Enterprise-wide Visibility & Enforcement
Verifiable Authenticity throughout SDLC
Integrated Threat & Vulnerability Detection
Software Transparency
PKI Support
Product and Enterprise Security
Auditors, Risk, & Compliance

- A software lifecycle security platform that:
  - Unifies Dev, PKI support, security, and compliance teams across the enterprise
  - Provides a single pane of glass for visibility
  - Enforces configurable product security controls across the enterprise
  - Integrates and automates deep threat & vulnerability scanning with secure authenticity controls (signing) into build workflows
  - Generates comprehensive software bills of materials
  - Easy and unobtrusive for dev teams to use
  - Easily scales across the enterprise

DIGICERT SOFTWARE TRUST MANAGER

Protecting the software development lifecycle from supply chain attacks

Enterprise Secure Code Signing

- Trusted public certificates
- Secure FIPS 140-2 key storage
- Automated certificate & key management

Enterprise Software Security

- Centralized security policy enforcement
- Integrated threat & vulnerability detection
- Policy-driven software release

Enterprise Software Transparency

- Enterprise-wide visibility for compliance
- Software bill of materials
- Irrefutable record of signing activities

Embedded software | Enterprise software | Cloud native software
Windows | Linux | macOS | iOS | Android | Kubernetes

- **PROTECTS** against software supply chain attacks
- **REDUCES RISKS** of releasing compromised software
- **INCREASES EFFICIENCY** of software, security, and compliance teams

SUMMARY

DIGICERT SOFTWARE TRUST MANAGER

Threat detection, secured code signing & SBOMs in a unified security workflow that is integrated, fast, easy, and automated for developers

Threat Detection

25B+ threat database

Complete binary scanning

Low impact to CI/CD

Secure Code Signing

Secured private keys

Role-based access control

Enterprise-wide visibility and signing policy

SBOMs

Deep binary decomposition

3rd party & open source

Regulatory compliance