AppSec, SCRM, and SBOM

: What they are and Why to care

Trac Bannon
Senior Principal
Software Architect & Digital Transformation Advisor
MITRE Advanced Software Innovation Center

June 2023

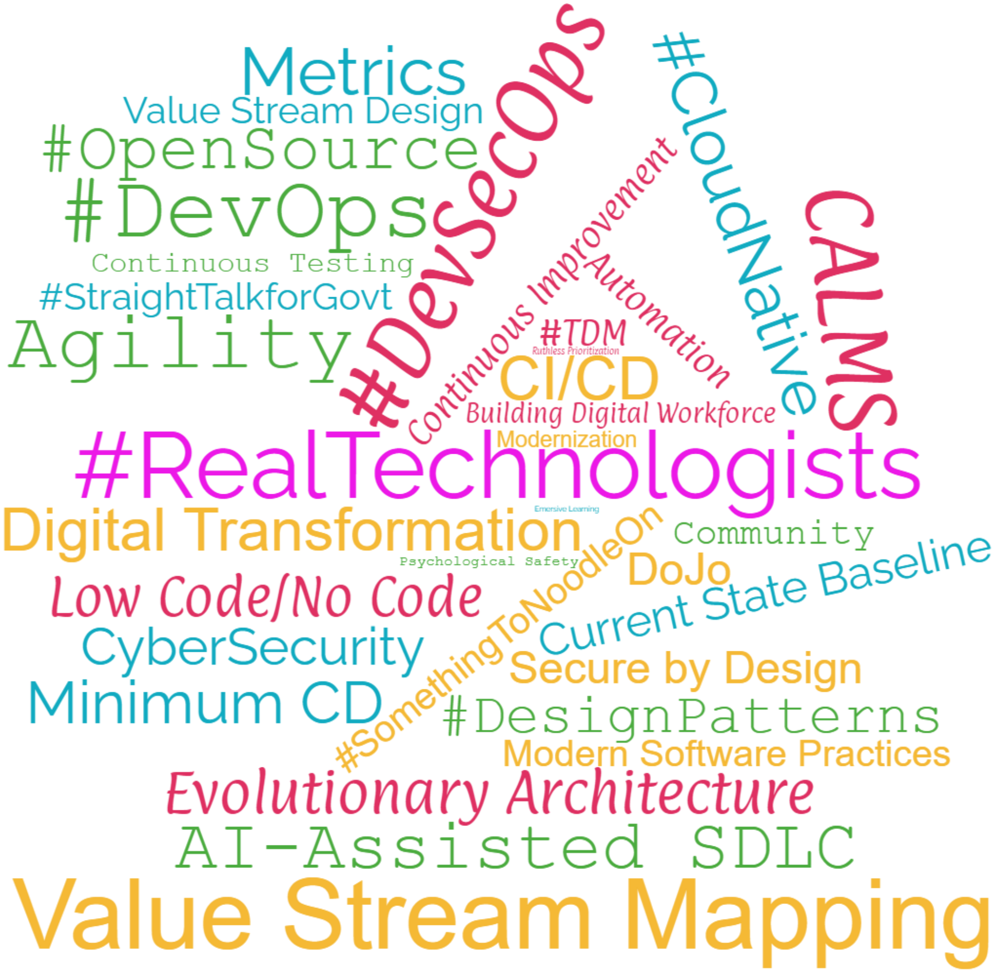
Who Am I?

/trās/

Tracy L. Bannon

"Trac"

Software architect | engineer | mentor | community leader



vsm consortium Influencer

Ambassador DevOps Institute Advancing the Humans of DevOps

IEEE Senior Member

MITRE

The Challenge

The **increased velocity** required to remain competitive in today's market together with the constant advances in **complexity and aggressiveness of cyber attackers** requires a new approach to remaining secure throughout the software development lifecycle.

The Solution

- Implementing **security checks** throughout your CI/CD pipelines
- **Continuously validating** the security and trustworthiness of the software supply chain
- Leveraging **next generation dynamic security scanning** techniques powered by AI

These steps are required to remain secure in the era of cloud scale.

MITRE

Key Considerations

Today's Modern Software Must be Secure by Design

"Secure by Design products are those where the ***security*** *of the customers* ***is a core business requirement, not just a technical feature.*** *Secure by Design principles should be implemented during the design phase of a product's development lifecycle to dramatically reduce the number of exploitable flaws before they are introduced to the market for broad use or consumption."*

Start with Secure Software Development Framework (SSDF)

https://www.cisa.gov/securebydesign

The way it was…

- Security scanning of your releases was an isolated event that happened at a specific stage of your release cycle
- Relied on static means to automatically analyze your code
- Humans evaluating the security of your implementation

The way it is…

- The old model does not scale
- Increasing reliance on open-sources packages further exposes releases to the vulnerabilities

Application Security And Software Supply Chain

AppSec

- measures and countermeasures taken during the software development lifecycle to protect applications
- identify, fix and prevent security vulnerabilities at all stages of the software lifecycle

Techniques:

- static application security testing (SAST)
- dynamic application security testing (DAST),
- interactive application security testing (IAST),
- software composition analysis (SCA)
- penetration testing.

Software Supply Chain

- part of the broader supply chain security and focuses on the security of all the components involved in the creation, delivery, and maintenance of a software product
- not just about the security of the code, but also about the systems and processes that involved in producing and maintaining that software.
- This includes third-party libraries, open-source software, development tools, container images, etc.

Traditional security scanning techniques are no longer sufficient.

AppSec and Software Supply Chain - Similarities

Goal: Both aim to maintain the integrity, confidentiality, and availability of the software and its data.

Threats: Both need to address a range of threats, from accidental vulnerabilities introduced during development to deliberate attacks aimed at compromising the software.

AppSec and Software Supply Chain - Differences

Scope: Securing the software itself versus securing all components and processes involved in creating, delivering, and maintaining the software.

Threat Model: Threats come from outside attackers versus threats from insiders or compromised elements within the supply chain.

Solution: Secure coding practices, security testing tools, and by patching software in a timely manner versus vetting third-party suppliers, using signed and reproducible builds, monitoring for unusual activity in the supply chain, etc.

Attack Vector: Application's own code or user data versus any element of the supply chain, such as a compromised third-party library or tool.

Core Principles of Application Security

MITRE

A Dozen AppSec Principles

1. Least Privilege
2. Defense in Depth
3. Secure by Default
4. Fail Securely
5. Separation of Duties
6. Least Common Mechanism
7. Security by Obscurity is not Security
8. Input Validation
9. Keep Security Simple
10. Patch and Update Regularly
11. Principle of Least Astonishment
12. Incident Response

Core Principles of Software Supply Chain Security

A Dozen SCRM Principles

1. Least Privilege Access
2. Verify Third-Party Components
3. Use of Signed Components
4. Reproducible Builds
5. Secure Your CI/CD Pipeline
6. Regular Audits
7. Patch and Update Regularly
8. Implement Strong Access Control
9. Defense in Depth
10. Incident Response
11. Transparency
12. Automate Security Checks

Can DevSecOps Principles Help?

Integrate security checks from development to deployment

Ensure your software supply chain is secure

Build releases that can't be tampered with

MITRE

1 - Integrate security checks

There are limitations of the traditional model of security scanning as an isolated event

Continuous security throughout the development lifecycle is imperative

We must implement security checks in CI/CD pipelines

Continuously validate the security and trustworthiness of software for greatest benefit

2- Ensure software supply chain is secure

There are risks associated with open-source packages and unknown codebases

It is important to secure and validate the software supply chain

We must apply techniques and tools to ensure the security of the supply chain

Emerging AI-driven alternatives can enhance software supply chain security

3 - Build releases that can't be tampered with

Tampering with software releases carries risk

We must ensure the integrity of releases

There are multiples techniques and best practices for building tamper-resistant releases

Incorporate security measures into the release process

What about the SBOM??

- Software Bill of Materials (SBOM) is essentially a list of components in a piece of software
- Incorporating SBOM into DevSecOps pipeline is prudent
  - ✓Automated Generation of SBOM
  - ✓Integrate SBOM in CI/CD Pipeline
  - ✓Version Control SBOMs
  - ✓Review and Audit SBOMs
  - ✓Vulnerability Scanning:

What and Why…

What about the How??

Tracy L. Bannon

tbannon@mitre.org

in https://www.linkedin.com/in/tracylbannon

@TracyBannon

https://tracybannon.tech

MITRE

SOLVING PROBLEMS
FOR A SAFER WORLD®

Disclaimer: The views, opinions and/or findings contained in this report are those of The MITRE Corporation and should not be construed as an official government position, policy, or decision, unless designated by other documentation.