Carnegie Mellon University
Software Engineering Institute

A Framework for Managing Security Risk across the Lifecycle and Supply Chain

AUGUST 23, 2023

Christopher Alberts
Principal Cybersecurity Analyst

Document Markings

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM23-0858

Agenda

- Background
- Acquisition Security Framework (ASF)
- Software Bill of Materials (SBOM) Framework
- Next Steps

A Framework for Managing Security Risk across the Lifecycle and Supply Chain

Background

Carnegie Mellon University
Software Engineering Institute

Barriers to Effective Management of Risk and Resilience

Carnegie Mellon University Software Engineering Institute

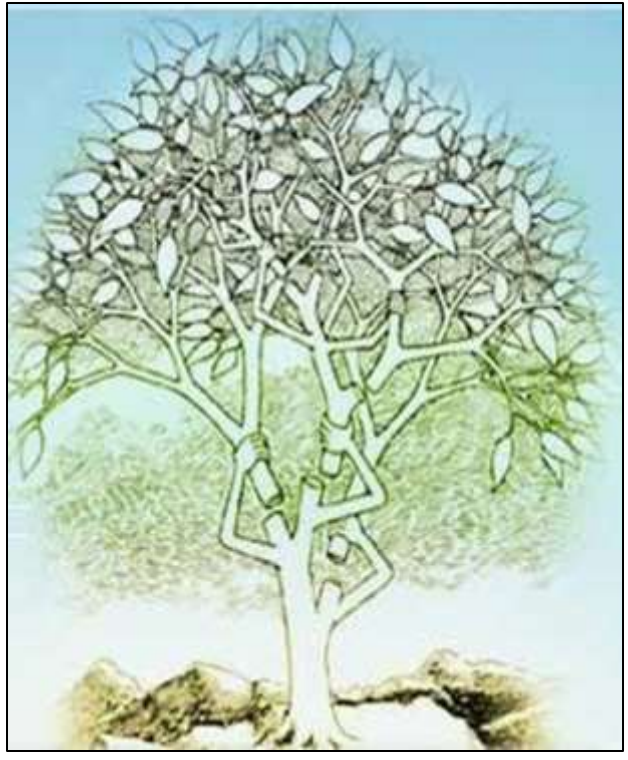
Complexity

Siloed departments operating under different poorly defined and managed requirements

- Engineering
- Supplier/acquisition
- Operations

Reliance on outdated technical/management approaches

Lack of coordinated planning, measurement, and reporting across the system lifecycle



Increasing Cyber Risks Require Attention

Carnegie Mellon University Software Engineering Institute

Growing role of third-party and open-source components using software-intensive solutions

Software-driven system-of-systems environments are becoming the norm.

Leadership roles and coordination points change and evolve throughout the lifecycle and lack integration with security engineering.

Coordinated measurement, reporting, and management is critical for addressing cyber challenges.

Software is Everywhere

You think you're building (or buying, or using) a product such as:

car or truck	satellite	mobile phone	development tools
home security system	aircraft	pacemaker	security tools
home appliance	financial system	bullets for a gun	

Actually you're getting ***a software platform:***

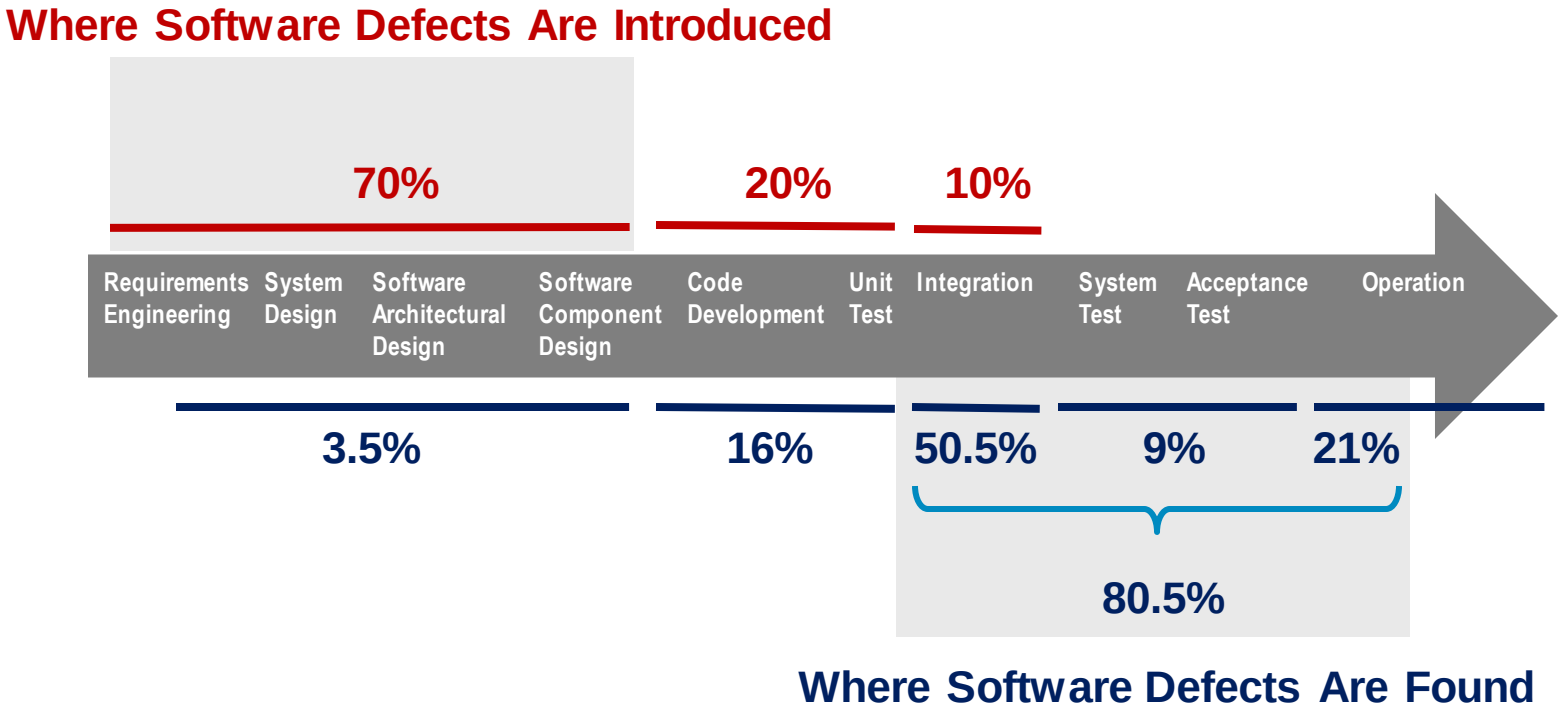
- Software is a part of almost everything we use.
- Software defines and delivers component and system communication.
- Software is used to build, analyze and secure software.

All software has defects:

- Best-in-class code has <600 defects per million lines of code (MLOC).
- Good code has around 1000 defects per MLOC.
- Average code has around 6000 defects per MLOC.

(based on Capers Jones research http://www.namcook.com/Working-srm-Examples.html)

Most Software Defects Are Found Long After They Are Introduced



Sources: *Critical Code*; NIST, NASA, INCOSE, and Aircraft Industry Studies

Supply Chain/Acquisition Risk is Increasing

More than 230,000 organizations were examined to discover their relationships with third parties. 98% of organizations have a relationship with a third party that has been breached within the last two years.
https://www.securityweek.com/98-of-firms-have-a-supply-chain-relationship-that-has-been-breached-analysis/

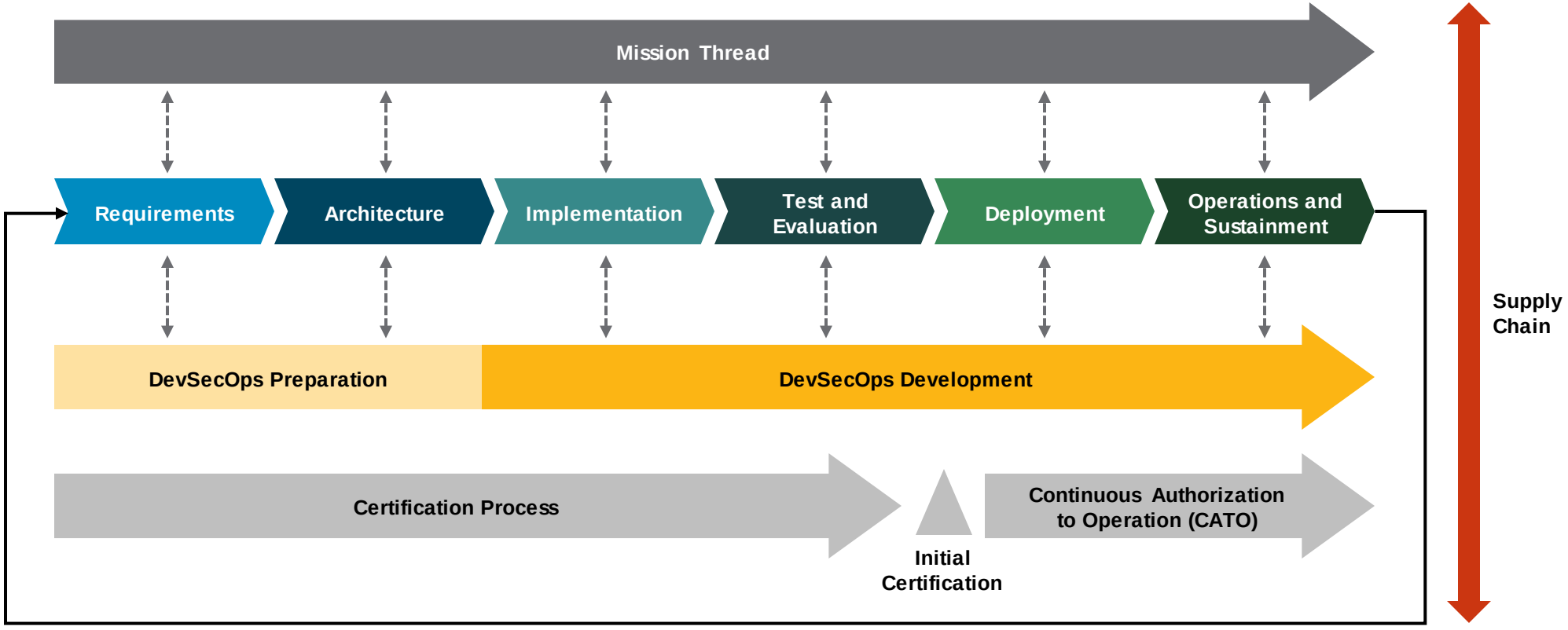
- Heartland Payment Systems (2009)
- Silverpop (2010)
- Epsilon (2011)
- New York State Electric and Gas (2012)
- Target (2013)
- Lowes (2014)
- AT&T(2014)
- HAVEX / Dragonfly attacks on energy industry (2014)
- DOD TRANSCOM contractor breaches (2014)
- Equifax (2017)
- Marriott (2018)
- SolarWinds (2020)
- Log4j (2021)
- Medibank (2022)
- ?...(2023)

A Framework for Managing Security Risk across the Lifecycle and Supply Chain

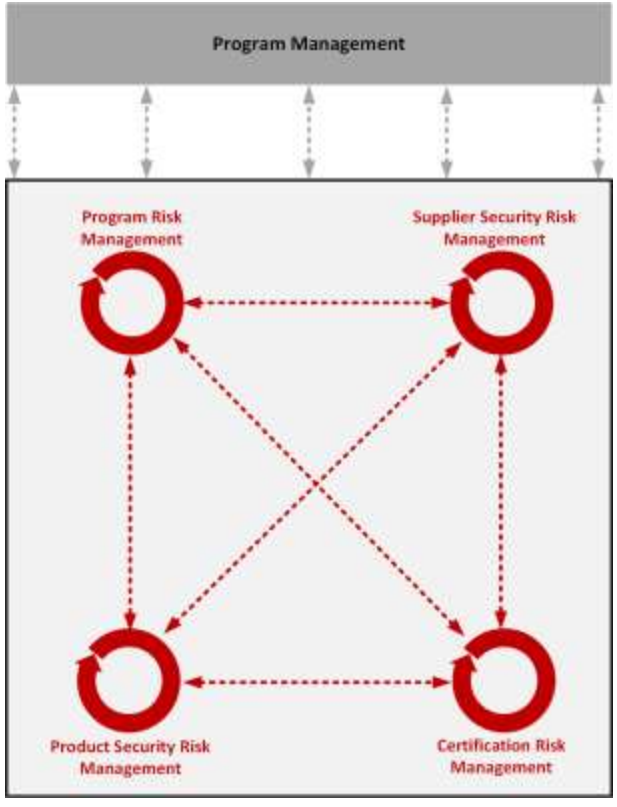
Acquisition Security Framework (ASF)

Carnegie Mellon University
Software Engineering Institute

Acquisition Cybersecurity Problem Space



Challenge: Integrated Security and Supplier Risk Management across the Organization



Security and supplier risk management are typically outside of the program risk management.

Information is dispersed across many documents, such as

- Program Protection Plan (PPP)
- Cybersecurity Plan
- Supply Chain Risk Management Plan

Cyber risk management activities must be

- Addressed collaboratively across the lifecycle and supply chain
- Integrated with program risk management

Challenge: Process Management and Improvement

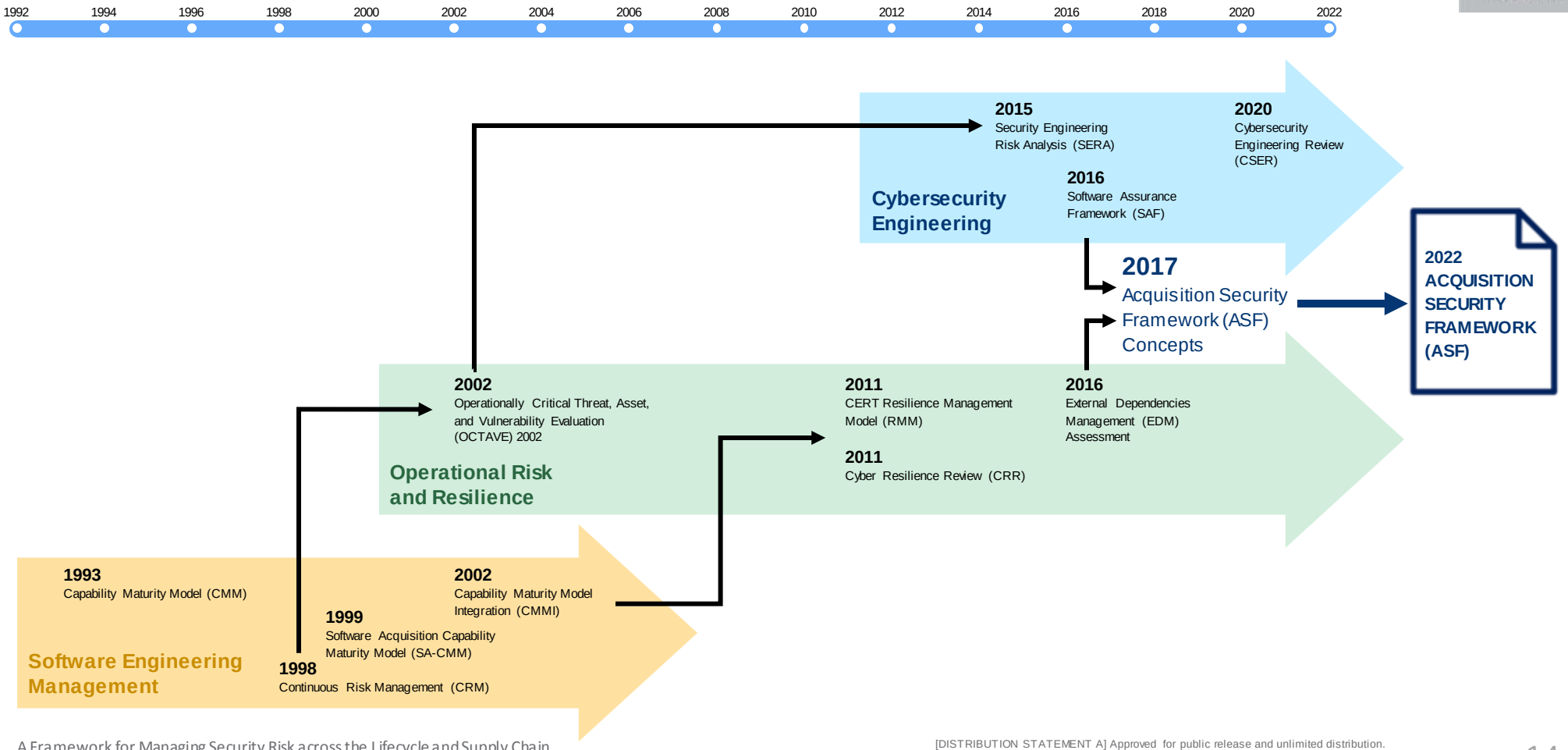
Higher degrees of process management translate to more stable environments that

- Produce consistent results over time
- Are able to achieve their missions during times of stress

The challenge is to implement an appropriate level of maturity for security practices across

- Multiple organizations/program units
- All lifecycle activities

ASF Research Lineage



What is the Acquisition Security Framework (ASF)?

Carnegie Mellon University Software Engineering Institute

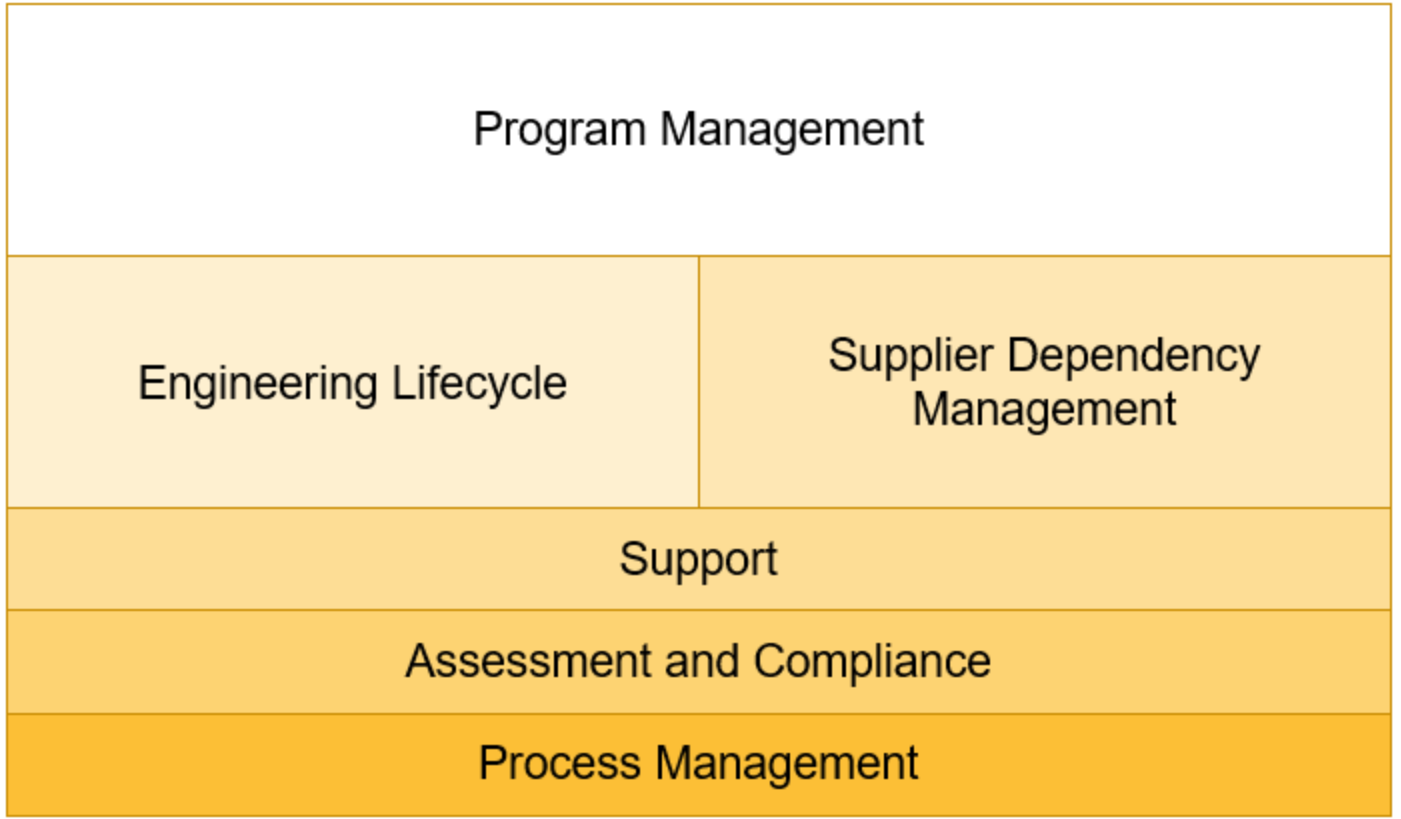
A collection of leading practices for building and operating secure and resilient software-reliant systems

Designed to enable system security and resilience engineering across the lifecycle and supply chain

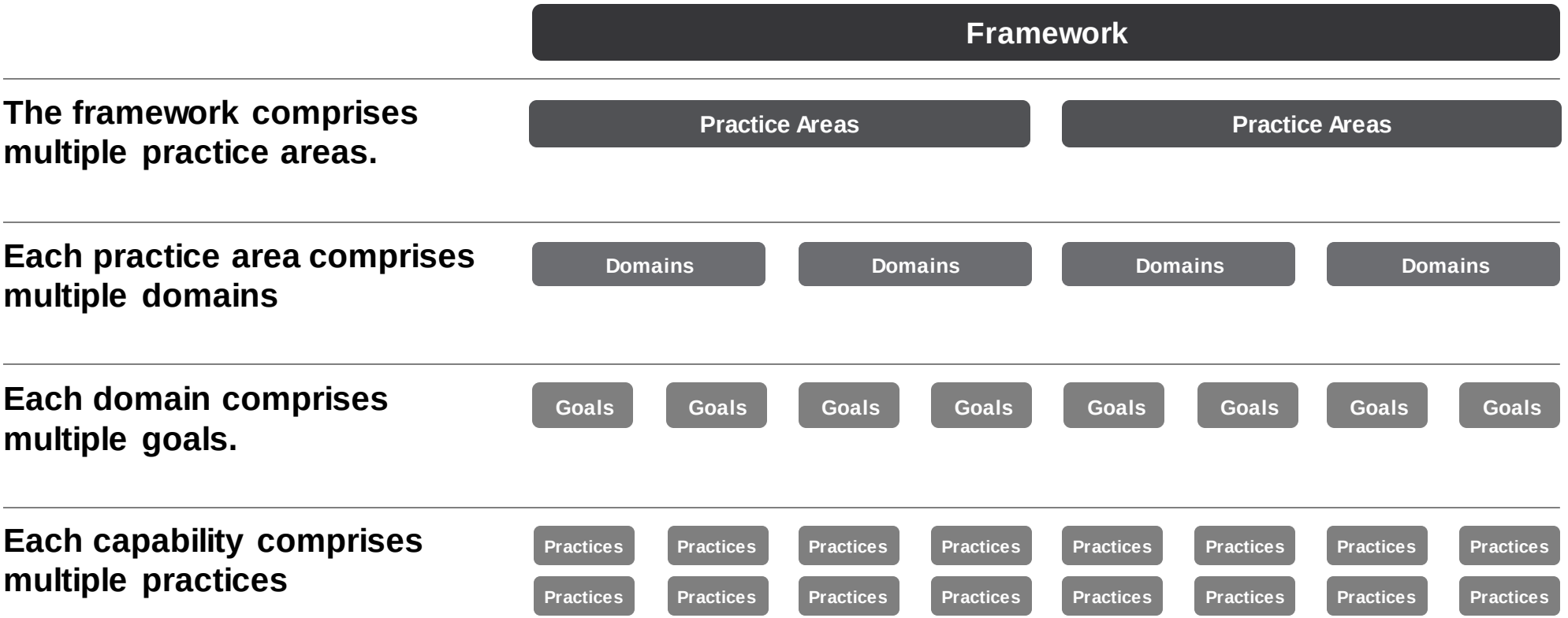
Provides a roadmap for building security and resilience into a system rather than attempting to “bolt them on” after deployment

Facilitates efficient and predictable systems environments and more manageable delivery and risk outcomes

ASF Practice Areas



ASF Structure



ASF Practice Areas -1

Practice Area	Areas of Focus
Program Management	Program planning, monitoring, and management Program requirements and risk management
Engineering Lifecycle	Engineering infrastructure, including development, test, and training environments Engineering management, including contractor management Engineering activities • Requirements • Architecture • Third-party components • Test and evaluation • Deployment • Operations and sustainment

ASF Practice Areas -2

Practice Area	Areas of Focus
Supplier Dependency Management	Forming relationships with suppliers, including contracting activities Managing supplier relationships Protection and sustainment activities performed by suppliers
Support	Program support, including configuration and change management Enterprise security support
Assessment and Compliance	Independent technical assessments Authorization to operate (ATO) Specialty engineering risk analysis (e.g., safety, nuclear surety)
Process Management	Strategic management of security and resilience across a program

ASF: Integrated Management

The ASF integrates two important technical perspectives:

- Engineering Lifecycle
- Supplier Dependency Management

Other areas facilitate a program's ability to manage engineering and supplier activities:

- Program Management
- Support
- Assessment and Compliance
- Process Management

ASF: Architecture Goal

Architecture

Goal 2—Security/resilience risks resulting from the architecture and design are identified and mitigated. The purpose of this goal is to identify and mitigate security/resilience risks resulting from the system's architecture and detailed design.
1. Is a security/resilience risk analysis of the architecture and detailed design performed?
2. Are identified security/resilience risks from the architecture and detailed design managed and tracked?
3. Is an architecture tradeoff analysis of quality attributes, including security/resilience, performed?
4. Are security/resilience risks resulting from architecture tradeoffs communicated to stakeholders?
5. Is the architecture's attack surface minimized based on the results of an attack-path analysis?
6. Is a cross check of the architecture and detailed design performed to resolve any issues or inconsistencies in security/resilience features?
7. Are security/resilience requirements updated periodically to reflect security/resilience changes to the architecture and detailed design?
8. Are reviews conducted with stakeholders to ensure that security/resilience risks resulting from the architecture and detailed design are mitigated sufficiently?

ASF: Third-Party Components Goal

Carnegie Mellon University Software Engineering Institute

Third-Party Components

Goal 3—Security/resilience risks that can affect third-party components (TPCs) are identified and mitigated. The purpose of this goal is to develop a bill of materials (BOM) for a product and ensure that operational security/resilience risks in the third-party software, firmware, and hardware are managed over time.
1. Are engineering relationships with third parties based on standards, guidelines, and policies?
2. Is a scheme that uniquely identifies each third-party component (TPC) implemented?
3. Is a repository to track TPC use in products implemented and maintained?
4. Are TPCs that are used in products identified and documented to create a bill of materials (BOM)?
5. Are suppliers evaluated and selected based on their use of secure/resilient development practices?
6. Is each TPC's operational risk assessed?
7. Are TPCs monitored for vulnerabilities and available updates?
8. Are TPCs prioritized for patch application based on operational risk?

ASF: Implementation Goal

Implementation

Goal 4—Vulnerabilities in software code are identified, managed, and tracked. The purpose of this goal is to identify and address vulnerabilities and security/resilience issues in the code base.	
1.	Is an appropriate suite of security/resilience tools integrated into the software development environment?
2.	Are secure coding standards applied?
3.	Are code reviews (e.g., peer reviews) performed to identify weaknesses and vulnerabilities?
4.	Is source code in critical components analyzed using white-box testing (e.g., static code analysis) during coding and unit testing to identify weaknesses and vulnerabilities?
5.	Is software in critical components analyzed using black-box testing (e.g., dynamic code analysis, vulnerability scanning) during integration testing to identify weaknesses and vulnerabilities?
6.	Are coding weaknesses and vulnerabilities remediated and tracked to resolution?

ASF: Supplier Performance Management Goal

Supplier Performance Management

Goal 2—Supplier performance is governed and managed. The purpose of this goal is to assess whether performance is considered when evaluating suppliers that support the security/resilience of the program or system.	
1.	Is the performance of suppliers monitored against the security/resilience requirements of the program or system?
2.	Is the responsibility for monitoring and managing the supplier established and maintained?
3.	Are supplier performance issues documented and reported to the appropriate stakeholders?
4.	Are corrective actions taken to address issues with supplier performance?
5.	Are corrective actions evaluated to ensure issues are remedied?

Key Lesson Learned from ASF Development

One practice framework will not address all problems that a program will face.

- Each practice framework is designed to address a specific problem.

Implement a “toolbox” approach.

- Develop a family of frameworks, each focused on a specific problem space.
  - Acquisition security
  - Software bill of materials (SBOM)
  - Zero Trust
  - Others

A Framework for Managing Security Risk across the Lifecycle and Supply Chain

Software Bill of Materials (SBOM) Framework

Carnegie Mellon University
Software Engineering Institute

What is an SBOM?

An SBOM is a formal record containing the details and supply chain relationships of various components used in building software.[1]

SBOMs are mandated under a federal directive EO 14028, Executive Order on Improving the Nation's Cybersecurity.[2]

1 *The Minimum Essential Elements of a Software Bill of Materials*, United States Department of Commerce, July 12, 2021, https://www.ntia.doc.gov/files/ntia/publications/sbom_minimum_elements_report.pdf

2 *Executive Order on Improving the Nation's Cybersecurity*, White House, 12 May 2021, https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/

SBOM Use Cases

Common SBOM use cases:[1]

- Build an SBOM for a system
- Receive and manage third-party SBOMs
- Manage known vulnerabilities
- Manage software versions
- Manage code reuse
- Manage software components that reach end of life
- Manage software licenses

1. National Telecommunications and Information Administration (NTIA), Use Cases Working Group, 2019.

SBOM Framework: Setting the Scope

Carnegie Mellon University Software Engineering Institute

To set the scope, we developed a scenario (based on use cases) for implementing an SBOM that includes :

- Develop / construct an SBOM.
- Use the SBOM to support identification of known vulnerabilities and risk reduction.

SBOM practices were established based on this scenario.

SBOM Framework

Goal	Focus
Requirements	Requirements for integrating SBOMs with the program's security/resilience activities are identified and managed.
Planning	A plan for using SBOMs to manage software security/resilience risks is developed.
Construction	Accurate and complete SBOM data is created for the system, subsystems, and components.
Vulnerability Management	Vulnerabilities are identified and managed in SBOM software components, leading to reduced system risk.
Management and support	Accurate, complete, and timely SBOM data is available to effectively manage risk.
Infrastructure	SBOM practices, software, and tools are integrated into the program's infrastructure.

Next Steps

Carnegie Mellon University
Software Engineering Institute

Next Steps

Refine and pilot the ASF and the SBOM Framework.

Develop additional frameworks to address customer needs:

- Security engineering (in progress)
  - Detailed practice guidance
  - Implementation details for selected practices
- Zero Trust (planned)

Publications

White Paper - Acquisition Security Framework (ASF): An Acquisition and Supplier Perspective on Managing Software-Intensive Systems' Cybersecurity Risk
https://resources.sei.cmu.edu/library/asset-view.cfm?assetID=887698

Technical Note - Alberts, Christopher; Bandor, Michael; Wallen, Charles; & Woody, Carol. Acquisition Security Framework (ASF): Managing Systems Cybersecurity Risk. CMU/SEI-2022-TN-003. Software Engineering Institute, Carnegie Mellon University. 2022.
http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=889215

White Paper - Software Bill of Materials Framework: Leveraging SBOMs for Risk Reduction
https://resources.sei.cmu.edu/library/asset-view.cfm?assetID=887698

Contact Information

Christopher Alberts
Principal Cybersecurity Analyst

Telephone: +1 412.268.3045
Email: cja@sei.cmu.edu