4-2017

Prototype Software Assurance Framework (SAF): Introduction and Overview

Christopher J. Alberts
Carnegie Mellon University, cja@cert.org

Carol Woody
Carnegie Mellon University, cwoody@cert.org

Software Engineering Institute
Carnegie Mellon University

Prototype Software Assurance Framework (SAF): Introduction and Overview

Christopher Alberts
Carol Woody

April 2017

TECHNICAL NOTE
CMU/SEI-2017-TN-001

CERT Division

http://www.sei.cmu.edu

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

This report was prepared for the
SEI Administrative Agent
AFLCMC/PZM
20 Schilling Circle, Bldg. 1305, 3rd floor
Hanscom AFB, MA 01731-2125

DM-0004590

Table of Contents

List of Figures

List of Tables

Acknowledgments

We thank Kris Britton of the National Security Agency (NSA) Center for Assured Software (CAS) for providing the funding to write this report. We thank Audrey Dorofee of the SEI for her work on developing Version 0.1 of the Software Assurance Framework (SAF). We also acknowledge the technical contributions of the following SEI technical staff members: Mary Ann Lapham, Julie Cohen, and Fred Schenker. They provided insights about how to integrate cybersecurity into the acquisition lifecycle and reviewed early versions of the SAF.

Abstract

Software is a growing component of modern business- and mission-critical systems. As organizations become more dependent on software, security-related risks to their organizational missions also increase. Traditional security-engineering approaches rely on addressing security risks during the operation and maintenance of software-reliant systems. The costs required to control security risks increase significantly when organizations wait until systems are deployed to address those risks. Field experiences of technical staff at the Software Engineering Institute (SEI) indicate that few programs currently implement effective cybersecurity practices early in the acquisition lifecycle. Recent DoD directives are beginning to shift programs' priorities regarding cybersecurity. As a result, researchers from the CERT Division of the SEI have started cataloging the cybersecurity practices needed to acquire, engineer, and field software-reliant systems that are acceptably secure.

This report introduces the prototype Software Assurance Framework (SAF), a collection of cybersecurity practices that programs can apply across the acquisition lifecycle and supply chain. The SAF can be used to assess an acquisition program's current cybersecurity practices and chart a course for improvement, ultimately reducing the cybersecurity risk of deployed software-reliant systems. This report presents Version 0.2 of the SAF and features three pilot applications of it.

Introduction

Software is a growing component of modern business- and mission-critical systems. As organizations become more dependent on software-driven technology, security-related risks to their organizational missions also increase. Traditional security engineering approaches rely on addressing security risks during the operation and maintenance of software-reliant systems. However, the costs required to mitigate software security risks increase significantly when organizations wait until systems are deployed to address those risks. It is more cost effective to address software security risks as early in the acquisition lifecycle as possible.

In November 2014, a group of researchers from the CERT Division at Carnegie Mellon University's Software Engineering Institute (SEI) started documenting cybersecurity[1] practices across the acquisition lifecycle in support of a gap analysis that we were asked to perform. The goal of the analysis was to identify gaps in current and planned software assurance services offered by a Department of Defense (DoD) service provider. To conduct the analysis, we needed a point of reference against which to evaluate the organization's services. A search of the literature did not yield a satisfactory framework that we could use to perform the gap analysis. As a result, we assumed the task of developing a prototype version of the Software Assurance Framework (SAF) that would serve as the basis for conducting the gap analysis.

The SAF defines a set of cybersecurity practices that programs should apply across the acquisition lifecycle and supply chain. The SAF can be used to assess a program's current cybersecurity practices and chart a course for improvement. By improving a program's cybersecurity practices, the SAF helps to (1) establish confidence in the program's ability to acquire software-reliant systems that are secure, and (2) reduce the cybersecurity risk of deployed software-reliant systems. When developing the SAF, we leveraged the software acquisition and cybersecurity expertise of the SEI's technical staff and referenced a variety of acquisition, development, process improvement, and cybersecurity documents, such as

- National Institute of Standards and Technology (NIST) Special Publication 800-53, titled *Security and Privacy Controls for Federal Information Systems and Organizations* [NIST 2013]
- NIST Special Publication 800-37, titled *Guide for Applying the Risk Management Framework to Federal Information Systems: A Security Life Cycle Approach* [NIST 2010]
- Department of Defense Instruction (DoDI) 5000-2, titled *Operation of the Defense Acquisition System* [DoDI 2003]
- Capability Maturity Model Integration (CMMI) [CMMI 2007]
- Build Security In Maturity Model (BSIMM) [BSIMM 2015]

We designated the prototype version of the SAF as SAF, v0.1. Version 0.1 uses the Defense Acquisition Management Framework (defined in the DoD's *Operation of the Defense Acquisition System* [DoDI 2003]) as its main organizing structure. However, as we started working with more

[1] The terms *security* and *cybersecurity* are used interchangeably in this document.

organizations, we quickly realized that those organizations often used unique lifecycle models. In January 2016, we initiated an effort to create a lifecycle-independent version of the SAF, which we called SAF, v0.2. This report documents the SAF, v0.2.

This latest version of the SAF (i.e., v0.2) can be tailored to DoD, federal, and industry lifecycle models as needed, making it more broadly applicable across sectors than the initial prototype. However, we want to stress that we consider the SAF, v0.2 to be a working prototype rather than a completed body of research. This report presents the structure and practices embodied in the SAF, v0.2.

The main goals of this report are to (1) raise awareness of the SAF in the software assurance and cybersecurity communities, and (2) initiate a dialogue with practitioners in those communities for refining and transitioning this work. In the next section, we begin the dialog by highlighting the importance of software security from a lifecycle perspective.

Importance of Software Security

Software assurance is defined as a level of confidence that software will function as intended and will be free of vulnerabilities, either intentionally or unintentionally designed or inserted as part of the software, throughout the acquisition lifecycle [NIA 2010]. Software assurance was legislatively mandated for the DoD in the National Defense Authorization Act for Fiscal Year 2013 [NDAA 2013]. The pursuit of software assurance is a worthy goal that must be translated into practical methods that acquirers, engineers, designers, and developers can apply throughout the acquisition lifecycle.

Software assurance is becoming increasingly important to organizations across all sectors because of software's increasing influence in business- and mission-critical systems. For example, consider how the size of flight software[2] has increased over the years. Between 1960 and 2000, the degree of functionality provided by software to the pilots of military aircraft has increased from 8% to 80%. At the same time, the size of software in military aircraft has grown from 1,000 lines of code in the F-4A to 1.7 million lines of code in the F-22. This trend is expected to continue over time [NASA 2009]. As software exerts more control over complex systems, like military aircraft, the potential risk posed by cybersecurity vulnerabilities will increase in kind.

Cost is another dimension of cybersecurity vulnerabilities that must be taken into account. Many cybersecurity vulnerabilities are considered to be software faults because their root causes can be traced to the software's requirements, architecture, design, or code. Studies have shown that the cost of addressing a software fault increases significantly (up to 200 times) if it is corrected during operations as opposed to design [Mainstay 2010, Microsoft 2014, Soo Hoo 2001]. In addition, rework related to defects consumes more than 50% of the effort associated with a software project. It is thus more cost effective to address software faults early in acquisition lifecycle rather than wait until operations. This principle applies to many operational security vulnerabilities as well.

[2] Flight software is a type of embedded real-time software used in avionics.

Operational security vulnerabilities generally have three main causes: (1) design weaknesses,[3] (2) implementation/coding errors, and (3) system configuration errors. Addressing design weaknesses as soon as possible is especially important because these weaknesses are not corrected easily after a system has been deployed. For example, software maintenance organizations normally cannot issue a patch to correct a fundamental security issue related to the software's requirements, architecture, or design. Remediation of design weaknesses normally requires extensive changes to the system; these changes can be costly and often prove to be impractical for the implemented system. As a result, software-reliant systems with design weaknesses often are allowed to operate under a higher degree of residual security risk, putting their associated operational missions in jeopardy.

Secure coding and operational security practices help address implementation/coding vulnerabilities and system configuration errors respectively. However, design weaknesses represent 19 of the top 25 weaknesses documented in the Common Weakness Enumeration[4] (CWE) [MITRE 2011]. The importance of design weaknesses in managing cybersecurity risk cannot be overstated.

Our field experience indicates that few acquisition and development programs currently implement effective cybersecurity practices. These programs have historically emphasized meeting performance, cost, and schedule objectives over meeting cybersecurity objectives. However, recent DoD directives promote a shift in programs' priorities. For example, the DoD issued an instruction mandating adherence to the NIST Risk Management Framework (RMF) for all DoD information technology programs [DoDI 2014]. As a result, DoD acquisition and development programs must pay more attention to cybersecurity. We developed the SAF for DoD programs to use as a touchstone for assessing and improving their cybersecurity practices. In the next section, we present the core structure of the SAF, v0.2.

SAF Structure

Figure 1 depicts Version 0.2 of the SAF, which defines cybersecurity practices for the following four categories:[5]

1. Process Management
2. Project Management
3. Engineering
4. Support

Each category comprises multiple areas of cybersecurity practice. In all, we have defined 19 practice areas for the SAF, v0.2. In addition, we have documented a set of cybersecurity practices for each area. The SAF features 76 cybersecurity practices across the 19 practice areas.

---

[3] In this report, we define a *design weakness* as a security-related defect in software's requirements, architecture, or design.

[4] MITRE maintains the Common Weakness Enumeration (CWE), an online dictionary of weaknesses that have been found in computer software. The purpose of the CWE is to facilitate the effective use of tools that identify, find, and resolve bugs, vulnerabilities, and exposures in computer software before the programs are publicly distributed or sold.

[5] The four categories of the SAF are aligned with the categories defined for CMMI process areas [CMMI 2007].

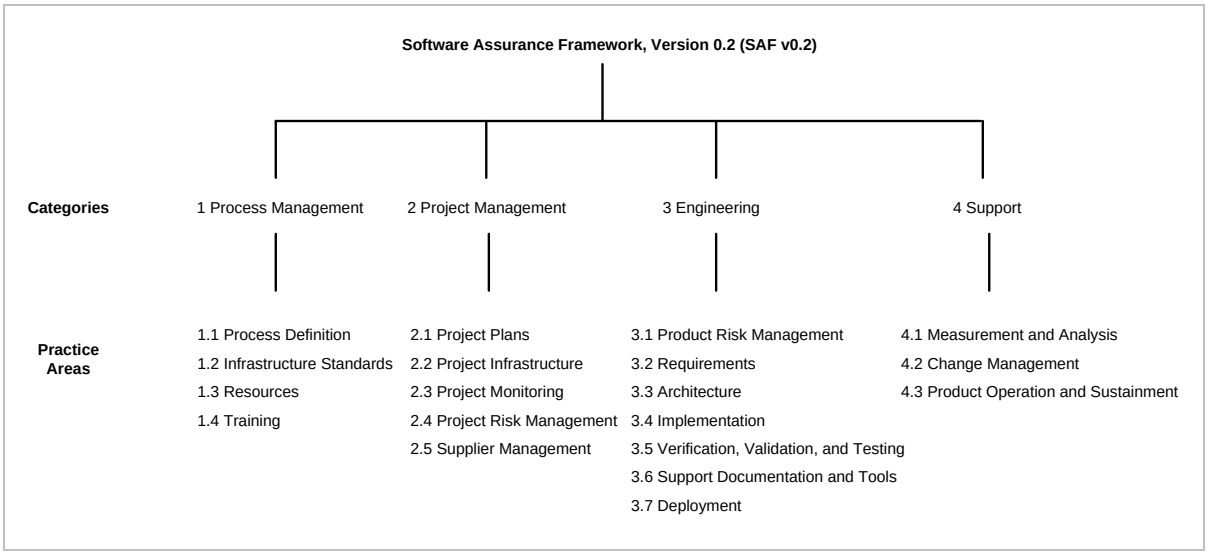

Figure 1: SAF, v0.2 Categories and Practice Areas

Table 1 highlights the SAF structure used for documenting cybersecurity practices. The table lists two SAF practices and their associated artifacts. Practice 1.1.1 is taken from Process Definition (SAF Area 1.1), while Practice 2.1.1 is one of the practices found under Project Plans (SAF Area 2.1).

Table 1: Example Cybersecurity Practices and Artifacts

Practice		**Artifacts**
1.1.1	Establish and maintain a standard set of cybersecurity policies, laws, and regulations with which projects must comply.	Organizational Cybersecurity Policies
2.1.1	Define and document cybersecurity objectives.	Program Plan Technology Development Strategy (TDS) Acquisition Strategy System Engineering Plan (SEP)

Some artifacts documented in the SAF are specific to cybersecurity. For example, in Table 1 the following artifact is listed for SAF Practice 1.1.1: Organizational Cybersecurity Policies. Here, an analyst is directed to examine the program's *cybersecurity polices* for evidence that SAF Practice 1.1.1 is implemented.

In contrast, some artifacts are generic and not specific to cybersecurity. For example, in Table 1 the following artifacts are listed for SAF Practice 2.1.1:

- Program Plan
- Technology Development Strategy (TDS)
- Acquisition Strategy
- System Engineering Plan (SEP)

For SAF Practice 2.1.1, an analyst is pointed to *generic program documentation* (e.g., program plan, TDS, acquisition strategy, SEP) for evidence that SAF Practice 2.1.1 is implemented. Although the program management documents are not focused specifically on cybersecurity, they should contain evidence of cybersecurity activities being performed by program personnel.

The Structure of this Report

The outline of this report is built on the structure of the SAF. Sections 1 through 4 provide the core technical content of the report. These four sections document cybersecurity practices for each of the SAF categories. The remainder of this section provides an overview of this report's audience, outline, and content.

This report presents our initial research and development related to the SAF. The primary audience for this report is someone seeking information about how to build security into software-reliant systems. Members of the audience for this report include software engineers, systems engineers, and system/software engineering managers. As we mature the SAF, we will continue to develop publications and products that are oriented toward this audience.

This report provides a conceptual framework of cybersecurity practices that can be applied across the acquisition lifecycle and supply chain and presents examples from our early piloting of the framework. The rest of this document includes the following sections:

- *Section 1: Process Management (Category 1)* presents cybersecurity practices for the Process Management category of the SAF.
- *Section 2: Project Management (Category 2)* presents cybersecurity practices for the Project Management category of the SAF.
- *Section 3: Engineering (Category 3)* presents cybersecurity practices for the Engineering category of the SAF.
- *Section 4: Support (Category 4)* presents cybersecurity practices for the Support category of the SAF.
- *Section 5: Applying the SAF* describes three pilot applications of the SAF performed by SEI technical staff members.
- *Section 6: Summary* presents a summary of the report's key concepts.
- *Appendix: SAF, v 0.2* includes a summary of all SAF practices by category and practice area.

1 Process Management (Category 1)

Process Management includes activities for defining, planning, monitoring, and improving organizational processes [CMMI 2007]. This category of the SAF defines the following four cybersecurity practice areas:

1.1. Process Definition

1.2. Infrastructure Standards

1.3. Resources

1.4. Training

In this section, we present the cybersecurity practices for each area, beginning with Process Definition.

1.1 Process Definition (Area 1.1)

Process Definition emphasizes the importance of (1) developing and documenting a standard set of cybersecurity processes that align with applicable policies, laws, and regulations, and (2) providing guidance for tailoring those processes to specific projects. Table 2 contains the cybersecurity practices and associated artifacts for the Process Definition practice area.

Table 2: Process Definition Practices and Artifacts

Practice		Artifacts
1.1.1	Establish and maintain a standard set of cybersecurity policies, laws, and regulations with which projects must comply.	Organizational Cybersecurity Policies
1.1.2	Establish and maintain standard cybersecurity processes (including lifecycle models) that align with policies, laws, and regulations.	Organizational Cybersecurity Processes Organizational Cybersecurity Lifecycles
1.1.3	Establish and maintain tailoring criteria and guidelines for the organization's cybersecurity processes (including lifecycle models).	Organizational Cybersecurity Tailoring Criteria and Guidelines

1.2 Infrastructure Standards (Area 1.2)

Infrastructure Standards is the second practice area of Process Management. This area of the SAF defines practices for establishing and maintaining criteria that govern cyber and physical security for the project's parent organization. Table 3 contains the cybersecurity practices (cyber and physical) and associated artifacts for the Infrastructure Standards practice area.

Table 3: Infrastructure Standards Practices and Artifacts

Practice		Artifacts
1.2.1	Establish and maintain cybersecurity standards for information technology systems and networks.	Organizational Cybersecurity Standards
1.2.2	Establish and maintain physical security standards for physical work spaces and facilities.	Organizational Physical Security Standards

1.3 Resources (Area 1.3)

The third practice area of Process Management is Resources. As used within the SAF, resources are a supply of something (e.g., people, expertise, money, and data) that a project has and can use when needed. Examples of cybersecurity resources are

- cybersecurity process assets (e.g., procedures, tools)
- security-related intelligence data (e.g., attack data, vulnerabilities, design weaknesses, abuse/misuse cases, threats)
- security features, frameworks, and patterns
- guidance for classifying data
- specialized security experts to assist project personnel

Table 4 contains the cybersecurity practices and associated artifacts for the Resources practice area.

Table 4: Resources: Practices and Artifacts

Practice		Artifacts
1.3.1	Establish and maintain standard cybersecurity process assets (e.g., procedures, tools) that align with processes and maintain them in a repository.	Organizational Cybersecurity Process Assets Security Resource Repository
1.3.2	Collect and maintain security-related intelligence data (e.g., attack data, vulnerabilities, design weaknesses, abuse/misuse cases, threats).	Security-Related Intelligence Data
1.3.3	Develop and document security features, frameworks, and patterns.	Approved Security Features, Frameworks, and Patterns
1.3.4	Establish and maintain guidance for classifying data.	Data Management System
1.3.5	Provide specialized security experts to assist project personnel.	Security Roles and Responsibilities

1.4 Training (Area 1.4)

The Training area presents practices for administering a cybersecurity training program for a project's personnel, including vendors, contractors, and outsourced workers. Table 5 contains the security-related practices and associated artifacts for the Training practice area.

Table 5: Training Practices and Artifacts

Practice		Artifacts
1.4.1	Provide security awareness training for program personnel (including vendors, contractors, and outsourced workers).	Project Training Plan Training Products Vendor Contracts and Service Level Agreements
1.4.2	Provide role-based security training for technical staff (including vendors, contractors, and outsourced workers).	Project Training Plan Training Products Vendor Contracts and Service Level Agreements
1.4.3	Track completion of security training activities.	Program Status Reports

2 Project Management (Category 2)

The Project Management category includes activities for planning, monitoring, and controlling the project. This category includes the following five practice areas:

2.1. Project Plans

2.2. Project Infrastructure

2.3. Project Monitoring

2.4. Project Risk Management

2.5. Supplier Management

In this section, we present the cybersecurity practices for each area of Project Management, beginning with Project Plans.

2.1 Project Plans (Area 2.1)

Practices in the Project Plans area focus on making sure that cybersecurity tasks and resources are factored appropriately into the project's strategy and plans. Table 6 contains the cybersecurity practices and associated artifacts for the Project Plans practice area.

Table 6: Project Plans Practices and Artifacts

Practice		Artifacts
2.1.1	Define and document cybersecurity objectives.	Program Plan Technology Development Strategy (TDS) Acquisition Strategy System Engineering Plan (SEP)
2.1.2	Integrate security tasks into the project plan.	Program Plan System Engineering Plan (SEP) Information Support Plan (ISP) Capability Production Document (CPD)
2.1.3	Define and assign cybersecurity roles and responsibilities.	Program Plan System Engineering Plan (SEP) Information Support Plan (ISP)
2.1.4	Provide adequate resources to implement planned cybersecurity tasks.	Program Plan
2.1.5	Select and implement a secure software development lifecycle (SSDL).	Program Processes
2.1.6	Define and implement a project compliance initiative for cybersecurity.	Program Compliance Documents

2.2 Project Infrastructure (Area 2.2)

Project personnel rely on the project's information technology systems and networks as well as physical work spaces and facilities to complete their assigned tasks. Threat actors may target a project's systems and networks to affect the confidentiality, integrity, or availability of project information. Similarly, some threat actors, such as malicious insiders, could use physical access to affect the confidentiality, integrity, or availability of project information.

The Project Infrastructure area of the SAF focuses on establishing and maintaining both the cyber and physical security of the project. While project personnel are not usually responsible for configuring information technology and securing physical work spaces, they are responsible for communicating their security requirements to the parties responsible for cyber and physical security for the project. Table 7 contains the security-related practices (cyber and physical) and associated artifacts for the Project Infrastructure practice area.

Table 7: Project Infrastructure Practices and Artifacts

Practice		Artifacts
2.2.1	Establish and maintain the cybersecurity of the project's information technology systems and networks.	Project Cybersecurity Documentation
2.2.2	Establish and maintain the physical security of the project's physical work spaces and facilities.	Project Physical Security Documentation

2.3 Project Monitoring (Area 2.3)

Project Monitoring is concerned with tracking the progress of a project's cybersecurity tasks. Here, project personnel essentially assess the project's current status in relation to its planned status. Table 8 contains the cybersecurity practices and associated artifacts for the Project Monitoring practice area.

Table 8: Project Monitoring Practices and Artifacts

Practice		Artifacts
2.3.1	Monitor the progress of the project's cybersecurity tasks.	Program Status Reports
2.3.2	Monitor project compliance with cybersecurity policies, laws, and regulations.	Program Compliance Documents
2.3.3	Conduct independent cybersecurity reviews of project tasks.	Independent Review Results

2.4 Project Risk Management (Area 2.4)

The Project Risk Management area, as defined in the SAF, focuses on identifying and managing project-level cybersecurity risks, such as risks related to cybersecurity resources and funding. In the SAF, project risk management is considered to be a *management* discipline. The project manager is the key stakeholder for project risk management. The Engineering category of the SAF (Category 3) includes an area titled *Product Risk Management*. The SAF differentiates product risk management from project risk management. Product risk management is considered to be an *engineering* discipline focused on a detailed analysis of cybersecurity risk in relation to the product's requirements, architecture, and design. The project's chief engineer is the key stakeholder for product risk management. Table 9 contains the security-related practices and associated artifacts for Project Risk Management.

Table 9: Project Risk Management Practices and Artifacts

Practice		Artifacts
2.4.1	Ensure that project strategies and plans address project-level cybersecurity risks (e.g., program risks related to cybersecurity resources and funding).	Program Plan Technology Development Strategy (TDS) Analysis of Alternatives (AoA)
2.4.2	Identify and manage project-level cybersecurity risks (e.g., program risks related to cybersecurity resources and funding).	Risk Management Plan Risk Repository

2.5 Supplier Management (Area 2.5)

Supplier Management requires project personnel to include cybersecurity considerations (e.g., risks, compliance requirements) into the project's oversight of contractors, suppliers, and vendors. Table 10 contains the security-related practices and associated artifacts for the Supplier Management practice area.

Table 10: Supplier Management Practices and Artifacts

Practice		Artifacts
2.5.1	Integrate cybersecurity considerations (e.g., risks, compliance requirements) into the proposal process.	Acquisition Strategy Request for Proposal (RFP) Statement of Work (SOW) Software Development Plan (SDP) Integrated Master Plan (IMP)
2.5.2	Define cybersecurity requirements for suppliers.	Acquisition Strategy Request for Proposal (RFP) Statement of Work (SOW) Service Level Agreement (SLA)

Practice		Artifacts
2.5.3	Select suppliers based on their ability to meet specified cybersecurity requirements.	Source Selection Criteria
2.5.4	Provide oversight of cybersecurity activities that are performed by suppliers.	Program Management Documentation
2.5.5	Conduct independent cybersecurity reviews of tasks being performed by suppliers.	Independent Review Results
2.5.6	Evaluate supplier deliverables against cybersecurity acceptance criteria.	Supplier Deliverables

3 Engineering (Category 3)

The Engineering category defines practices for building security into software-reliant systems. The main objective of Engineering is to integrate cybersecurity practices into a project's software and systems engineering activities. This category of the SAF features the following seven practice areas:

3.1. Product Risk Management

3.2. Requirements

3.3. Architecture

3.4. Implementation

3.5. Verification, Validation, and Testing

3.6. Support Documentation and Tools

3.7. Deployment

In this section, we present the cybersecurity practices for each area of Engineering, beginning with Product Risk Management.

3.1 Product Risk Management (Area 3.1)

The Product Risk Management area focuses on the detailed analysis of cybersecurity risk in relation to the product's requirements, architecture, and design. In the SAF, product risk management is considered to be an *engineering* activity. The project's chief engineer is the key stakeholder for product risk management. The *Project Management* category of the SAF (Category 2) includes an area titled *Project Risk Management*. The SAF differentiates project risk management from product risk management. Project risk management is considered to be a *management* discipline focused on identifying and managing project-level cybersecurity risks, such as risks related to cybersecurity resources and funding. The project manager is the key stakeholder for project risk management. Table 11 contains the cybersecurity practices and associated artifacts for the Product Risk Management practice area.

Table 11: Product Risk Management Practices and Artifacts

Practice		Artifacts
3.1.1	Perform a basic cybersecurity risk analysis (e.g., health check) of all systems/components (including custom-developed software, commercial-off-the-shelf software, and open source software) to establish their criticality.	Risk Management Plan Risk Repository
3.1.2	Perform a deep-dive cybersecurity risk analysis (e.g., threat modeling, NIST Risk Management Framework) of critical systems/components.	Risk Management Plan Risk Repository System Threat Assessment (STAR)
3.1.3	Document the cybersecurity controls needed to protect critical systems/components.	Program Protection Plan (PPP)
3.1.4	Implement cybersecurity controls needed to protect critical systems/components.	Engineering Documents

3.2 Requirements (Area 3.2)

A requirement is a statement that documents a necessary attribute, capability, characteristic, or quality of a system that provides utility to stakeholders. A security requirement specifies a security capability or need that must be satisfied by a system. Requirements analysis should determine which security needs or capabilities the system should provide. The purpose of the Requirements area of the SAF is to produce, analyze, and manage security requirements for the customer, product, and product components. Table 12 contains the cybersecurity practices and associated artifacts for the Requirements practice area.

Table 12: Requirements Practices and Artifacts

Practice		Artifacts
3.2.1	Define and document cybersecurity requirements.	Concept of Operations (CONOPS) Initial Capabilities Document (ICD) Capability Development Document (CDD) Technical Requirements Document (TRD)
3.2.2	Conduct formal reviews of cybersecurity requirements.	System Requirements Review (SRR)

3.3 Architecture (Area 3.3)

The process for developing a software product ultimately includes two design phases that may overlap in their execution: (1) preliminary design and (2) detailed design. Preliminary design establishes a product's capabilities and defines the product's architecture, which typically includes product partitions, product components, system states, and both internal and external interfaces [CMMI 2007]. The detailed design defines the structure and capabilities of product components and interfaces [CMMI 2007]. The Architecture area of the SAF identifies cybersecurity practices for both phases. Table 13 contains the cybersecurity practices and associated artifacts for the Architecture practice area.

Table 13: Architecture Practices and Artifacts

Practice		Artifacts
3.3.1	Perform cybersecurity risk analysis of the architecture.	System and Software Architecture Descriptions Functional Architecture
3.3.2	Incorporate cybersecurity controls into the architecture.	System and Software Architecture Descriptions Functional Architecture

Practice		Artifacts
3.3.3	Conduct formal reviews of the cybersecurity controls in the architecture.	Preliminary Design Review (PDR)
3.3.4	Perform cybersecurity risk analysis of the design.	System and Software Architecture Descriptions Detailed Design/Physical Architecture
3.3.5	Incorporate cybersecurity controls into the design.	Software Design Description
3.3.6	Conduct formal reviews of the cybersecurity aspects of the design.	Critical Design Review (CDR)

3.4 Implementation (Area 3.4)

During system implementation, engineers construct system elements that meet stakeholder and system requirements. For software, code is developed and integrated during the implementation phase. Secure coding practices, peer reviews, and application of code analysis tools are important aspects of identifying vulnerabilities and cybersecurity issues in the code base. Table 14 contains the cybersecurity practices and associated artifacts for the Implementation practice area.

Table 14: Implementation Practices and Artifacts

Practice		Artifacts
3.4.1	Apply secure coding principles.	Secure Coding Standards
3.4.2	Conduct code reviews (e.g., peer reviews) of selected components to identify coding vulnerabilities.	Code Review Results
3.4.3	Analyze selected components using source code analysis tools to identify coding vulnerabilities.	Automated Code Review Tools and Results
3.4.4	Track and manage coding vulnerabilities.	Code Review Results Centralized Code Review Repository

3.5 Verification, Validation, and Testing (Area 3.5)

Verification is the process of ensuring that a system or component meets its specified requirements. For cybersecurity, verification focuses on ensuring that the security requirements have been met. Validation is the process of demonstrating that a system or component fulfills its intended use when placed in its intended environment. From a cybersecurity perspective, validation helps to ensure that a system or component will be able to fulfill its mission or gracefully degrade as planned when under attack from cyber threats. Independent verification and validation are normally performed by a person or group that is not part of the development team.

Software testing is an activity that provides stakeholders with information about the quality of the software being developed. Software testing provides an objective, independent view of the software program or application with the intent of finding software errors or other defects. Finding

vulnerabilities and security issues is an important aspect of security testing, which occurs at multiple points in the acquisition lifecycle.

Development test and evaluation is conducted throughout the acquisition process to (1) assist in engineering design and development, and (2) verify that technical performance specifications (e.g., security requirements) have been met. Operational test and evaluation is a fielded test of critical components or the integrated system under realistic conditions to determine operational effectiveness and operational suitability. Table 15 contains the cybersecurity practices and associated artifacts for the Verification, Validation, and Testing practice area.

Table 15: Verification, Validation, and Testing Practices and Artifacts

Practice		Artifacts
3.5.1	Develop cybersecurity test cases.	Test Cases
3.5.2	Conduct cybersecurity test readiness reviews.	Test Readiness Review Results
3.5.3	Perform functional and risk-based cybersecurity testing for selected components (unit testing of cybersecurity).	Test and Evaluation Master Plan (TEMP) Developmental Test and Evaluation (DT&E)
3.5.4	Perform functional and risk-based cybersecurity testing of the integrated system.	Test and Evaluation Master Plan (TEMP)
3.5.5	Perform operational security testing for the integrated system.	Test and Evaluation Master Plan (TEMP)
3.5.6	Perform independent cybersecurity validation of selected components.	Independent Validation Results
3.5.7	Perform independent cybersecurity verification of selected components.	Independent Verification Results

3.6 Support Documentation and Tools (Area 3.6)

Support documentation refers to security-related engineering information that is produced during the system and software engineering process. This information includes security plans, security risk and mitigation plans, security requirements, and security architecture documentation. Support tools include applications, programs, and software used to support the operation or maintenance of the system. Table 16 contains the cybersecurity practices and associated artifacts for the Support Documentation and Tools practice area.

Table 16: Support Documentation and Tools Practices and Artifacts

Practice		Artifacts
3.6.1	Compile relevant security-related engineering documentation for system administrators and users.	Engineering Documentation
3.6.2	Develop tools that support the secure operation of the system (e.g., setting a secure configuration and auditing against it).	Support Tools

Practice		Artifacts
3.6.3	Conduct formal reviews of security-related engineering documentation and support tools before releasing them to stakeholders.	Review Results

3.7 Deployment (Area 3.7)

Deployment is the activity where a system is installed, tested, and implemented in its production environment. Table 17 contains the cybersecurity practices and associated artifacts for the Deployment practice area.

Table 17: Deployment Practices and Artifacts

Practice		Artifacts
3.7.1	Obtain security sign off for system release.	Assessment and Authorization Plan
3.7.2	Obtain the authority to operate in a production environment (i.e., accept residual cybersecurity risk to operations).	Assessment and Authorization Plan
3.7.3	Protect the code during transport and installation.	Deployment Policy

4 Support (Category 4)

The Support category addresses activities that enable the development, operation, and sustainment of a product [CMMI 2007]. The Support category in the SAF comprises the following three practice areas:

4.1. Measurement and Analysis

4.2. Change Management

4.3. Product Operation and Sustainment

In this section, we describe the cybersecurity practices for each area of Support, beginning with Measurement and Analysis.

4.1 Measurement and Analysis (Area 4.1)

The objective of Measurement and Analysis is to develop and sustain a measurement capability that supports management's information needs for cybersecurity. Table 18 contains the cybersecurity practices and associated artifacts for the Measurement and Analysis practice area.

Table 18: Measurement and Analysis Practices and Artifacts

Practice		Artifacts
4.1.1	Define and improve cybersecurity measures.	Program Plan Program Status Reports
4.1.2	Collect and analyze cybersecurity measures.	Program Plan Program Status Reports

4.2 Change Management (Area 4.2)

The purpose of Change Management is to control changes to all cybersecurity configuration items (e.g., requirements specification, architecture documentation, code, user documents, and support tools). Table 19 contains the cybersecurity practices and associated artifacts for the Change Management practice area.

Table 19: Change Management Practices and Artifacts

Practice		Artifacts
4.2.1	Incorporate cybersecurity changes into the strategy and plan documents and artifacts.	Change Requests Configuration/Change Management System
4.2.2	Incorporate cybersecurity changes into the engineering documents and artifacts.	Change Requests Configuration/Change Management System

4.3 Product Operation and Sustainment (Area 4.3)

The final practice area of Support is Product Operation and Sustainment. Here, cybersecurity engineers provide technical support for the operation of deployed systems. Examples include

- cybersecurity risk analysis, assessment, and vulnerability scanning of operational systems
- penetration testing of software
- support of system- and network-monitoring activities and incident response as required

Table 20 contains the cybersecurity practices and associated artifacts for the Product Operation and Sustainment practice area.

Table 20: Product Operation and Sustainment Practices and Artifacts

Practice		Artifacts
4.3.1	Perform detailed cybersecurity risk analyses of operational systems.	Operational Risk Management Plan Operational Risk Repository
4.3.2	Assess cybersecurity during maintenance testing.	Maintenance Testing Results
4.3.3	Conduct periodic penetration testing of all software to identify cybersecurity vulnerabilities.	Penetration Testing Results
4.3.4	Conduct deep-dive penetration testing of critical software to identify cybersecurity vulnerabilities.	Penetration Testing Results
4.3.5	Run vulnerability scanning tools on operational systems.	Vulnerability Management Reports
4.3.6	Remediate identified cybersecurity vulnerabilities and risks.	Defect Management System
4.3.7	Monitor the behavior of operational software/systems to identify signs of attack.	Software Monitoring Results
4.3.8	Respond to cybersecurity incidents as appropriate.	Incident Response Ticketing System
4.3.9	Ensure the ability to roll back to a previous version of the system when needed and maintain the expected level of cybersecurity.	Configuration/Change Management System
4.3.10	Communicate suggested product changes or improvements related to cybersecurity to the engineering team.	Field Change Requests Configuration/Change Management System

5 Applying the SAF

To this point in this report, we have focused on describing the structure and content of the SAF. In this section, we turn our attention away from the content of the framework and examine how we have used the SAF to support our field work. In particular, we discuss how we used the SAF to support the following engagements:

- conducting a gap analysis of the software assurance services provided by an organization
- integrating software security practices into an organization's existing policies and procedures
- generating a candidate set of cybersecurity engineering metrics

In this section, we provide a brief summary of each engagement.

5.1 Gap Analysis

In the *Introduction* of this report, we explained how we developed the initial version of the SAF and used it as the basis for a gap analysis. The goal of the gap analysis was to identify gaps in current and planned software assurance services offered by a DoD service provider. We developed the SAF, v0.1, after our search of the applicable literature failed to find a satisfactory framework to serve as the basis for the gap analysis.

We used the Defense Acquisition Management Framework (defined in the DoD's *Operation of the Defense Acquisition System* [DoDI 2003]) as the organizing structure for the SAF, v0.1. Figure 2 provides a visual representation of the framework's nine practice areas.

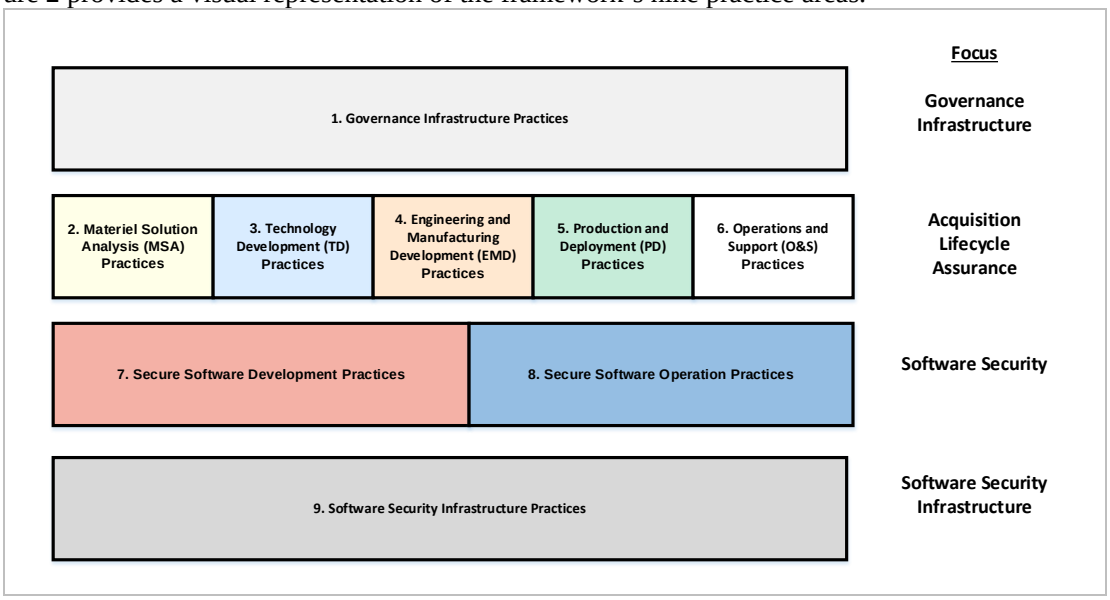

Figure 2: Framework of Assurance Practices: Nine Practice Areas

The following list summarizes each of the areas depicted in Figure 2:

1. *Governance Infrastructure*—foundational practices needed to establish and implement a program's assurance policies
2. *Materiel Solution Analysis (MSA)*—the assurance activities that must be performed during the MSA phase of the acquisition lifecycle
3. *Technology Development (TD)*—the assurance activities that must be performed during the TD phase of the acquisition lifecycle
4. *Engineering and Manufacturing Development (EMD)*—the assurance activities that must be performed during the EMD phase of the acquisition lifecycle
5. *Production and Deployment (PD)*—the assurance activities that must be performed during the PD phase of the acquisition lifecycle
6. *Operations and Support (O&S)*—the assurance activities that must be performed during the O&S phase of the acquisition lifecycle
7. *Secure Software Development*—the technical activities that must be performed when developing software with desired security characteristics
8. *Secure Software Operation*—the technical activities that must be performed when operating software in a secure manner
9. *Software Security Infrastructure*—foundational practices needed to develop and operate secure software

The first step in a gap analysis activity is to collect relevant data. We conducted interviews with personnel from the service provider to gather information about software assurance services currently offered by the organization. We then mapped the software assurance services provided by the organization to the appropriate practices in the SAF, v0.1. After completing the mapping, we identified gaps where the current services did not adequately address the practices to which they were mapped. We also identified high-priority services that the provider might consider offering in the future.

5.2 Process Improvement

Our second application of the SAF was with a Level 5 CMMI organization. The purpose of the engagement was to identify how the organization could integrate software security practices into its existing policies and procedures. For this engagement, we restructured the SAF based on CMMI's structure. We grouped the cybersecurity practices from the SAF, v0.1, into CMMI's four categories: (1) Process Management, (2) Project Management, (3) Engineering, and (4) Support.

Each category comprises multiple areas of cybersecurity practice. In all, we defined 19 practice areas for the SAF, v0.2, and documented cybersecurity practices for each area. The SAF features 76 cybersecurity practices across the 19 practice areas.

For our customer engagement, we reviewed the organization's current policies, procedures, and templates for the CMMI process areas in Table 21. After meeting with the organization's technical staff, we jointly determined that most of the organization's policies could remain unchanged since they were written from a sufficiently broad perspective. We did recommend that the organi-

zation consider creating a new policy for cybersecurity, which would identify roles and responsibilities for cybersecurity, define cybersecurity policy, and provide pointers to related cybersecurity procedures.

Table 21: CMMI Process Areas Analyzed

Category	CMMI Process Area
Process Management	Organizational Process Definition (OPD) Organizational Process Performance (OPP)
Project Management	Project Planning (PP) Project Monitoring and Control (PMC) Supplier Agreement Management (SAM) Risk Management (RSKM)
Engineering	Requirements Management (REQM) Requirements Development (RD) Technical Solution (TS) Product Integration (PI) Verification (VER) Validation (VAL)
Support	Configuration Management (CM) Measurement and Analysis (MA)

We also recommended that the organization decide whether to develop a separate policy for product risk management or update its existing risk management policy to include product risk management. As noted in Sections 2 and 3, the SAF differentiates project risk management from product risk management. Project Risk Management (Area 2.4) is considered to be a management discipline focused on identifying and managing project-level cybersecurity risks, such as risks related to cybersecurity resources and funding.

In contrast, Product Risk Management (Area 3.1) is considered to be an engineering activity. It focuses on *detailed* analyses of cybersecurity risk in relation to the product's requirements, architecture, and design. Our research and development activities in the area of risk management indicate that project and product risk management require different levels of analysis. We recommended that the organization address both types of risk management in its policies. Its current policy primarily addressed project risk management.

Based on our analysis of the organization's procedures and templates, we recommended several changes. In this report, we focus on one of our suggestions—updating the organization's Requirements Development (RD) procedures. Here, we suggested that the organization develop a separate procedure for developing security requirements. Methods for eliciting security requirements typically combine security risk analysis techniques with standard requirements specification techniques. For example, the SEI Security Quality Requirements Engineering (SQUARE) method defines a means for eliciting, categorizing, and prioritizing security requirements for software-reliant

systems and applications. SQUARE specifies a nine-step approach for developing security requirements [Allen 2008]:

1. Agree on definitions.
2. Identify assets and security goals.
3. Develop artifacts to support security requirements definition.
4. Perform a security risk assessment.
5. Select elicitation techniques.
6. Elicit security requirements.
7. Categorize security requirements as to their level (e.g., system, software) and whether they are requirements or other kinds of constraints.
8. Prioritize security requirements.
9. Inspect security requirements.

We recommended that the organization adopt an approach for specifying security requirements (either SQUARE or an equivalent) and develop a procedure based on the selected approach.

While we suggested that the organization create a stand-alone procedure for developing security requirements, we did not see the need to make changes to its existing procedure for Requirements Management (REQM). The procedure for Requirements Management was sufficiently broad and could be used to manage all types of stakeholder and technical requirements, including security requirements.

5.3 Metrics

For our third application of the SAF, we used version 0.2 to generate a candidate set of engineering metrics for a DoD organization. We used a standard software engineering method, Goal-Question-Metric (GQM), to identify a candidate set of engineering metrics [Basili 1984].

We worked with the organization's senior managers to identify the following organizational goal for software assurance: *Supply software to the user with acceptable software risk*. Using that goal and the definition of software assurance,[6] we identified two sub-goals:

- sub-goal 1: Supply software to the user that functions in the intended manner.
- sub-goal 2: Supply software to the user with a minimal number of exploitable vulnerabilities.

We then decided to focus on sub-goal 2 for our engagement with the organization. We used the SAF as the organizing structure for developing GQM questions. For example, we developed the following question for the Engineering category: *Do engineering activities minimize the potential for exploitable software vulnerabilities?*

[6] As explained in the *Introduction* of this report, *software assurance* is defined as a level of confidence that software functions as intended and is free of vulnerabilities, either intentionally or unintentionally designed or inserted as part of the software, throughout the lifecycle [NIA 2010].

We then developed a question for each practice area in the Engineering category of the SAF:

- Product Risk Management: *Does the program manage cybersecurity risk in software components?*
- Requirements: *Does the program manage software security requirements?*
- Architecture: *Does the program incorporate appropriate cybersecurity controls in its software architecture and design?*
- Implementation: *Does the program minimize the number of vulnerabilities inserted into the code?*
- Verification, Validation, and Testing: *Does the program test, validate, and verify cybersecurity in its software components?*
- Support Tools and Documentation: *Does the program develop tools and documentation to support the secure configuration and operation of software components?*
- Deployment: *Does the program consider cybersecurity during the deployment of software components?*

In this report, we provide an example of the candidate metrics that we developed for the Requirements area of the SAF. Table 22 contains the specific questions we generated for requirements and the candidate metrics we identified for each question.

Table 22: Candidate Measures/Metrics Mapped to Security Requirements Questions

Security Requirements Question		**Candidate Metrics**
[1]	Have software engineers received training in how to develop security requirements for software?	% of software engineers trained in security requirements development
[2]	Has a security risk analysis been conducted?	Number of software security risks controlled/mitigated (e.g., high and medium risks) Number of software security risks accepted/transferred Number of software security controls/mitigations selected for requirements development
[3]	Have software security requirements been defined and documented?	Traceability ▪ Number of selected controls/mitigations without corresponding security requirements ▪ Number of security requirements traced to high and medium risks
[4]	Do the software security requirements mitigate high-priority software security risks?	Traceability ▪ Number of selected controls/mitigations without corresponding security requirements ▪ Number of security requirements traced to high and medium risks

Security Requirements Question		Candidate Metrics
[5]	Have reviews (e.g., peer reviews, inspections, and independent reviews) of software security requirements been conducted?	Completeness ▪ Number of to-be-determined (TBD) and to-be-added (TBA) items for software security requirements Correctness ▪ Number of software security requirements not validated ▪ % of software security requirements that have not been validated Understandability ▪ Number of software security requirements not understood by reviewers
[6]	Are changes to software security requirements being managed?	Volatility ▪ Number of change requests for software security requirements ▪ % of software security requirements changed

This is an ongoing engagement. We are currently working with the organization to select an initial set of engineering metrics from the list of candidates. The organization will then begin using the selected metrics in its management and decision-making activities.

6 Summary

We began this report with the story about how we came to develop the SAF. We were asked to perform a gap analysis of the software assurance services offered by a DoD service provider, and we needed a point of reference against which to evaluate the organization's services. We first documented a set of cybersecurity practices and organized them around the activities in the Defense Acquisition Management Framework. Then we used the framework as the basis for conducting the gap analysis.

Because of the SAF's significant role in our field work, we decided to document the current version of the prototype SAF in this report. In Sections 1 through 4, we presented the structure and practices embodied in the SAF, v0.2.

In Section 5, we discussed another field activity where we analyzed how a Level 5 CMMI organization might integrate software security practices into its existing policies and procedures. For this engagement, we restructured the SAF based on CMMI's structure by grouping the cybersecurity practices from SAF, v0.1, into CMMI's four categories: (1) Process Management, (2) Project Management, (3) Engineering, and (4) Support. The result of this project was the SAF, v0.2. Section 5 also includes a summary of our recent metrics project, where we used the SAF, v0.2, to generate a candidate set of engineering metrics for a DoD organization.

The SAF proved to be a useful tool in the three engagements featured in this report. The framework provides us with a basis for describing, assessing, and measuring an acquisition program's cybersecurity practices across its lifecycle and supply chain. However, it is important to emphasize that we consider the SAF to be a working prototype. Each field application of the SAF has provided important insights about how we can improve the framework. We restructured the SAF after our initial gap analysis work; we plan to update it in the future based on our more recent field work.

Our goals when writing this report were to raise awareness of the SAF in the software assurance and cybersecurity communities and initiate a dialogue for refining and transitioning this work to practitioners in those communities. We do not consider the SAF to be a completed piece of work; rather, we view it as a "living" framework that we intend to nurture and grow in the years ahead. We see this report as the first step in raising awareness of our work and initiating a dialogue with the community.

Appendix SAF, v0.2

This appendix presents Version 0.2 of the SAF in its entirety. This version is a prototype that SEI technical staff members have used when working with customer organizations. Future versions will incorporate lessons learned from the SEI's field work related to cybersecurity engineering as well as feedback from the community. The SAF, v0.2, defines cybersecurity practices for the following four categories:

1. Process Management
2. Project Management
3. Engineering
4. Support

Each category comprises multiple areas of cybersecurity practice. In the SAF, a set of cybersecurity practices is defined for each area. In all, we defined 19 practice areas for the SAF, v0.2. In addition, we documented a set of cybersecurity practices for each area. The SAF features 76 cybersecurity practices across the 19 practice areas.

Finally, relevant acquisition and engineering artifacts are documented for each cybersecurity practice. Some artifacts specified in the SAF are specific to cybersecurity (e.g., cybersecurity policies, cyber-capable resources), while other artifacts are generic and not specific to cybersecurity (e.g., program planning documents, training databases). Analysts can look for evidence that a cybersecurity practice has been implemented by examining the artifacts related to that practice. The remainder of this appendix presents the cybersecurity practices featured in the SAF, v 0.2.

Category	Area	Practice	Artifacts
1 Process Management	1.1 Process Definition	1.1.1 Establish and maintain a standard set of cybersecurity policies, laws, and regulations with which projects must comply.	Organizational Cybersecurity Policies
		1.1.2 Establish and maintain standard cybersecurity processes (including lifecycle models) that align with policies, laws, and regulations.	Organizational Cybersecurity Processes Organizational Cybersecurity Lifecycles
		1.1.3 Establish and maintain tailoring criteria and guidelines for the organization's cybersecurity processes (including lifecycle models).	Organizational Cybersecurity Tailoring Criteria and Guidelines
	1.2 Infrastructure Standards	1.2.1 Establish and maintain cybersecurity standards for information technology systems and networks.	Organizational Cybersecurity Standards
		1.2.2 Establish and maintain physical security standards for physical work spaces and facilities.	Organizational Physical Security Standards
	1.3 Resources	1.3.1 Establish and maintain standard cybersecurity process assets (e.g., procedures, tools) that align with processes and maintain them in a repository.	Organizational Cybersecurity Process Assets Security Resource Repository
		1.3.2 Collect and maintain security-related intelligence data (e.g., attack data, vulnerabilities, design weaknesses, abuse/misuse cases, threats).	Security-Related Intelligence Data
		1.3.3 Develop and document security features, frameworks, and patterns.	Approved Security Features, Frameworks, and Patterns
		1.3.4 Establish and maintain guidance for classifying data.	Data Management System
		1.3.5 Provide specialized security experts to assist project personnel.	Security Roles and Responsibilities

Category	Area	Practice	Artifacts
	1.4 Training	1.4.1 Provide security awareness training for program personnel (including vendors, contractors, and outsources workers).	Project Training Plan Training Products Vendor Contracts and Service Level Agreements
		1.4.2 Provide role-based security training for technical staff (including vendors, contractors, and outsources workers).	Project Training Plan Training Products Vendor Contracts and Service Level Agreements
		1.4.3 Track completion of security training activities.	Program Status Reports
2 Project Management	2.1 Project Plans	2.1.1 Define and document cybersecurity objectives.	Program Plan Technology Development Strategy (TDS) Acquisition Strategy System Engineering Plan (SEP)
		2.1.2 Integrate security tasks into the project plan.	Program Plan System Engineering Plan (SEP) Information Support Plan (ISP) Capability Production Document (CPD)
		2.1.3 Define and assign cybersecurity roles and responsibilities.	Program Plan System Engineering Plan (SEP) Information Support Plan (ISP)
		2.1.4 Provide adequate resources to implement planned cybersecurity tasks.	Program Plan
		2.1.5 Select and implement a secure software development lifecycle (SSDL).	Program Processes
		2.1.6 Define and implement a project compliance initiative for cybersecurity.	Program Compliance Documents

Category	Area	Practice	Artifacts
	2.2 Project Infrastructure	2.2.1 Establish and maintain the cybersecurity of the project's information technology systems and networks.	Project Cybersecurity Documentation
		2.2.2 Establish and maintain the physical security of the project's physical work spaces and facilities.	Project Physical Security Documentation
	2.3 Project Monitoring	2.3.1 Monitor the progress of the project's cybersecurity tasks.	Program Status Reports
		2.3.2 Monitor project compliance with cybersecurity policies, laws, and regulations.	Program Compliance Documents
		2.3.3 Conduct independent cybersecurity reviews of project tasks.	Independent Review Results
	2.4 Project Risk Management	2.4.1 Ensure that project strategies and plans address project-level cybersecurity risks (e.g., program risks related to cybersecurity resources and funding).	Program Plan Technology Development Strategy (TDS) Analysis of Alternatives (AoA)
		2.4.2 Identify and manage project-level cybersecurity risks (e.g., program risks related to cybersecurity resources and funding).	Risk Management Plan Risk Repository
	2.5 Supplier Management	2.5.1 Integrate cybersecurity considerations (e.g., risks, compliance requirements) into the proposal process.	Acquisition Strategy Request for Proposal (RFP) Statement of Work (SOW) Software Development Plan (SDP) Integrated Master Plan (IMP)
		2.5.2 Define cybersecurity requirements for suppliers.	Acquisition Strategy Request for Proposal (RFP) Statement of Work (SOW) Service Level Agreement (SLA)
		2.5.3 Select suppliers based on their ability to meet specified cybersecurity requirements.	Source Selection Criteria
		2.5.4 Provide oversight of cybersecurity activities that are performed by suppliers.	Program Management Documentation

Category	Area	Practice	Artifacts
		2.5.5 Conduct independent cybersecurity reviews of tasks being performed by suppliers.	Independent Review Results
		2.5.6 Evaluate supplier deliverables against cybersecurity acceptance criteria.	Supplier Deliverables
3 Engineering	3.1 Product Risk Management	3.1.1 Perform a basic cybersecurity risk analysis (e.g., health check) of all systems/components (including custom-developed software, commercial-off-the-shelf software, and open source software) to establish their criticality.	Risk Management Plan Risk Repository
		3.1.2 Perform a deep-dive cybersecurity risk analysis (e.g., threat modeling, NIST Risk Management Framework) of critical systems/components.	Risk Management Plan Risk Repository System Threat Assessment (STAR)
		3.1.3 Document the cybersecurity controls needed to protect critical systems/components.	Program Protection Plan (PPP)
		3.1.4 Implement cybersecurity controls needed to protect critical systems/components.	Engineering Documents
	3.2 Requirements	3.2.1 Define and document cybersecurity requirements.	Concept of Operations (CONOPS) Initial Capabilities Document (ICD) Capability Development Document (CDD) Technical Requirements Document (TRD)
		3.2.2 Conduct formal reviews of cybersecurity requirements.	System Requirements Review (SRR)
	3.3 Architecture	3.3.1 Perform cybersecurity risk analysis of the architecture.	System and Software Architecture Descriptions Functional Architecture

Category	Area	Practice	Artifacts
		3.3.2 Incorporate cybersecurity controls into the architecture.	System and Software Architecture Descriptions Functional Architecture
		3.3.3 Conduct formal reviews of the cybersecurity controls in the architecture.	Preliminary Design Review (PDR)
		3.3.4 Perform cybersecurity risk analysis of the design.	System and Software Architecture Descriptions Detailed Design/Physical Architecture
		3.3.5 Incorporate cybersecurity controls into the design.	Software Design Description
		3.3.6 Conduct formal reviews of the cybersecurity aspects of the design.	Critical Design Review (CDR)
	3.4 Implementation	3.4.1 Apply secure coding principles.	Secure Coding Standards
		3.4.2 Conduct code reviews (e.g., peer reviews) of selected components to identify coding vulnerabilities.	Code Review Results
		3.4.3 Analyze selected components using source code analysis tools to identify coding vulnerabilities.	Automated Code Review Tools and Results
		3.4.4 Track and manage coding vulnerabilities.	Code Review Results Centralized Code Review Repository
	3.5 Verification, Validation, and Testing	3.5.1 Develop cybersecurity test cases.	Test Cases
		3.5.2 Conduct cybersecurity test readiness reviews.	Test Readiness Review Results
		3.5.3 Perform functional and risk-based cybersecurity testing for selected components (unit testing of cybersecurity).	Test and Evaluation Master Plan (TEMP) Developmental Test and Evaluation (DT&E)
		3.5.4 Perform functional and risk-based cybersecurity testing of the integrated system.	Test and Evaluation Master Plan (TEMP)
		3.5.5 Perform operational security testing for the integrated system.	Test and Evaluation Master Plan (TEMP)

Category	Area	Practice	Artifacts
		3.5.6 Perform independent cybersecurity validation of selected components.	Independent Validation Results
		3.5.7 Perform independent cybersecurity verification of selected components.	Independent Verification Results
	3.6 Support Documentation and Tools	3.6.1 Compile relevant security-related engineering documentation for system administrators and users.	Engineering Documentation
		3.6.2 Develop tools that support the secure operation of the system (e.g., setting a secure configuration and auditing against it).	Support Tools
		3.6.3 Conduct formal reviews of security-related engineering documentation and support tools before releasing them to stakeholders.	Review Results
	3.7 Deployment	3.7.1 Obtain security sign off for system release.	Assessment and Authorization Plan
		3.7.2 Obtain the authority to operate in a production environment (i.e., accept residual cybersecurity risk to operations).	Assessment and Authorization Plan
		3.7.3 Protect the code during transport and installation.	Deployment Policy
4 Support	4.1 Measurement and Analysis	4.1.1 Define and improve cybersecurity measures.	Program Plan Program Status Reports
		4.1.2 Collect and analyze cybersecurity measures.	Program Plan Program Status Reports
		4.1.3 Store cybersecurity measurement data appropriately.	Program Data Repository
	4.2 Change Management	4.2.1 Incorporate cybersecurity changes into the strategy and plan documents and artifacts.	Change Requests Configuration/Change Management System
		4.2.2 Incorporate cybersecurity changes into the engineering documents and artifacts.	Change Requests Configuration/Change Management System
	4.3 Product Operation and Sustainment	4.3.1 Perform detailed cybersecurity risk analyses of operational systems.	Operational Risk Management Plan Operational Risk Repository

Category	Area	Practice		Artifacts
		4.3.2	Assess cybersecurity during maintenance testing.	Maintenance Testing Results
		4.3.3	Conduct periodic penetration testing of all software to identify cybersecurity vulnerabilities.	Penetration Testing Results
		4.3.4	Conduct deep-dive penetration testing of critical software to identify cybersecurity vulnerabilities.	Penetration Testing Results
		4.3.5	Run vulnerability scanning tools on operational systems.	Vulnerability Management Reports
		4.3.6	Remediate identified cybersecurity vulnerabilities and risks.	Defect Management System
		4.3.7	Monitor the behavior of operational software/systems to identify signs of attack.	Software Monitoring Results
		4.3.8	Respond to cybersecurity incidents as appropriate.	Incident Response Ticketing System
		4.3.9	Ensure the ability to roll back to a previous version of the system when needed and maintain the expected level of cybersecurity.	Configuration/Change Management System
		4.3.10	Communicate suggested product changes or improvements related to cybersecurity to the engineering team.	Field Change Requests Configuration/Change Management System

References

URLs are valid as of the publication date of this document.

[Allen 2008]
Allen, Julia H.; Barnum, Sean; Ellison, Robert J.; McGraw, Gary; & Mead, Nancy R. *Software Security Engineering: A Guide for Project Managers*. Addison-Wesley. 2008.

[Basili 1984]
Basili, Victor R. & Weiss, David M. "A Methodology for Collecting Valid Software Engineering Data." *IEEE Transactions on Software Engineering.* Volume SE-10. Number 6. November 1984. Pages 728-738.

[BSIMM 2015]
McGraw, Gary; Migues, Sammy; & West, Jacob. *Building Security In Maturity Model (BSIMM) Version 6*. Cigital. 2015. https://www.bsimm.com/

[CMMI 2007]
Chrissis, Mary Beth; Konrad, Mike; & Shrum, Sandy. *CMMI Second Edition: Guidelines for Process Integration and Product Improvement.* Addison-Wesley, 2007.

[DoDI 2003]
Department of Defense. *Operation of the Defense Acquisition System* (DoDI 5000-2). Department of Defense, 2003. http://www.acq.osd.mil/dpap/Docs/new/5000.2%2005-12-06.pdf

[DoDI 2014]
Department of Defense. *Risk Management Framework (RMF) for DoD Information Technology (IT)*. DoDI 8510.01. Department of Defense. 2014. http://www.dtic.mil/whs/directives/corres/pdf/851001_2014.pdf

[Mainstay 2010]
Mainstay Partners. *Does Application Security Pay?* September 2010. http://h30528.www3.hp.com/Security/Fortify_Mainstay_ROI_Study.pdf

[Microsoft 2014]
Microsoft Corporation. *Benefits of the SDL.* September 2014. http://www.microsoft.com/security/sdl/about/benefits.aspx

[MITRE 2011]
MITRE Corporation. *2011 CWE/SANS Top 25 Most Dangerous Software Errors.* 2011. http://cwe.mitre.org/top25/

[NASA 2009]
National Aeronautics and Space Administration (NASA). *Final Report, NASA Study on Flight Software Complexity*. NASA Jet Propulsion Laboratory, Systems and Software Division. 2009. http://www.nasa.gov/pdf/418878main_FSWC_Final_Report.pdf

[NDAA 2013]
One Hundred Twelfth Congress of the United States of America. *National Defense Authorization Act for Fiscal Year 2013*. Washington, DC, 2013. http://www.gpo.gov/fdsys/pkg/BILLS-112hr4310enr/pdf/BILLS-112hr4310enr.pdf

[NIA 2010]
Committee on National Security Systems. *National Information Assurance (IA) Glossary CNSS Instruction*. CNSS Instruction No. 4009. 2010. http://www.dtic.mil/whs/directives/corres/pdf/851001_2014.pdf

[NIST 2010]
National Institute of Standards and Technology. *Guide for Applying the Risk Management Framework to Federal Information Systems: A Security Life Cycle Approach*. NIST Special Publication 800-37, Revision 1. National Institute of Standards and Technology. 2010. http://csrc.nist.gov/publications/nistpubs/800-37-rev1/sp800-37-rev1-final.pdf

[NIST 2013]
National Institute of Standards and Technology. *Security and Privacy Controls for Federal Information Systems and Organizations*. NIST Special Publication 800-53, Revision 4. National Institute of Standards and Technology. 2013. http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf

[Soo Hoo 2001]
Soo Hoo, K. S.; Sudbury, A. W.; & Jaquith, A. R. "Tangible ROI through Secure Software Engineering." *Secure Business Quarterly 1*, 2 (Fourth Quarter 2001).

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
(Leave Blank)	April 2017	Final

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
Prototype Software Assurance Framework (SAF): Introduction and Overview	FA8721-05-C-0003

6. AUTHOR(S)

Christopher Alberts & Carol Woody

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213	CMU/SEI-2017-TN-001

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFLCMC/PZE/Hanscom Enterprise Acquisition Division 20 Schilling Circle Building 1305 Hanscom AFB, MA 01731-2116	n/a

11. SUPPLEMENTARY NOTES

12A DISTRIBUTION/AVAILABILITY STATEMENT	12B DISTRIBUTION CODE
Unclassified/Unlimited, DTIC, NTIS	

13. ABSTRACT (MAXIMUM 200 WORDS)

Software is a growing component of modern business- and mission-critical systems. As organizations become more dependent on software, security-related risks to their organizational missions also increase. Traditional security-engineering approaches rely on addressing security risks during the operation and maintenance of software-reliant systems. The costs required to control security risks increase significantly when organizations wait until systems are deployed to address those risks. Field experiences of technical staff at the Software Engineering Institute (SEI) indicate that few programs currently implement effective cybersecurity practices early in the acquisition lifecycle. Recent DoD directives are beginning to shift programs' priorities regarding cybersecurity. As a result, researchers from the CERT Division of the SEI have started cataloging the cybersecurity practices needed to acquire, engineer, and field software-reliant systems that are acceptably secure.

This report introduces the prototype Software Assurance Framework (SAF), a collection of cybersecurity practices that programs can apply across the acquisition lifecycle and supply chain. The SAF can be used to assess an acquisition program's current cybersecurity practices and chart a course for improvement, ultimately reducing the cybersecurity risk of deployed software-reliant systems. This report presents Version 0.2 of the SAF and features three pilot applications of it.

14. SUBJECT TERMS	15. NUMBER OF PAGES
cybersecurity practices, acquisition lifecycle, software-reliant systems	46

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL