Carnegie Mellon University
Software Engineering Institute

Security Engineering Framework (SEF): Managing Security and Resilience Risks Across the Systems Lifecycle

Christopher Alberts
Charles M. Wallen
Carol Woody
Michael Bandor
Tom Merendino

December 2024

SPECIAL REPORT
CMU/SEI-2024-SR-022
DOI: 10.1184/R1/25029359

CERT® Division

https://www.sei.cmu.edu

Design: REV-03.18.2016.0 | Template: 02.22.2023

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.

DM24-0879

Table of Contents

List of Figures

Abstract

Software is a growing component of modern business- and mission-critical systems. As a result, software assurance is becoming increasingly important to organizations across all sectors. A key aspect of software assurance is keeping security and resilience risks within an acceptable tolerance across the systems lifecycle. The Security Engineering Framework (SEF) is a collection of software-focused engineering practices for managing security and resilience risks across the systems lifecycle. It provides a roadmap for building security and resilience into software-reliant systems and maintaining the system's security/resilience capabilities during operations and sustainment (O&S). SEF practices help ensure that engineering processes, software, and tools are secure and resilient, reducing the risk that attackers will disrupt program and system information and assets. Acquisition programs can use the SEF to assess their current security/resilience engineering practices and chart a course for improvement, ultimately reducing security/resilience risks in deployed software-reliant systems. The SEF organizes practices into a hierarchy of goals and domains and provides in-depth guidance for all goals and practices. SEF guidance describes the capability represented by each goal and provides an elaboration of each practice in the framework. This report provides a detailed description of the SEF, including its organizing structure, practices, and guidance.

Acknowledgements

We would like to thank Sandy Shrum and Barbara White of the SEI for editing and formatting the technical content of this report; David Biber for developing the report's graphics; and Ed Desautels for performing a final quality check. We would also like to thank SEI technical staff members who have contributed to the SEI's cybersecurity engineering body of knowledge over the years, most notably Audrey Dorofee, Robert Ellison, and Nancy Mead.

Preface

The Security Engineering Framework (SEF) is a collection of software-focused engineering practices for managing security and resilience risks across the systems lifecycle. It provides a roadmap for building security and resilience into software-reliant systems and maintaining the system's security and resilience capabilities during operations and sustainment (O&S). The SEF organizes practices into a hierarchy of domains and goals and provides in-depth guidance for all goals and practices.

This report provides a detailed description of the SEF, including its organizing structure, practices, and guidance. It comprises three parts: Part 1 presents overall concepts and terminology; Part 2 presents the SEF's guidance; and Part 3 contains appendices that include a list of acronyms, a glossary of terms and definitions, and a list of references.

Part 1: Introduction and Key Concepts

Part 1 sets the context for the SEF by examining core concepts related to systems engineering, software engineering, and risk management. The primary audience for Part 1 is anyone interested in learning about the SEF's foundational concepts.

Those interested in applying SEF goals and practices from Part 2 will find Part 1 useful because it defines the organizational and risk contexts for the SEF. Reading Part 1 before using Part 2 is highly recommended because it provides an overview of the general concepts and the SEF terminology covered in Part 2. This approach is especially important for risk-related content since there is no standard vocabulary for discussing risk. Readers may also need to translate SEF guidance to suit their organization's unique situations. Part 1 will help them adapt SEF guidance effectively.

Part 1 includes the following sections:

- **Section 1: Introduction** provides a brief introduction to the SEF and some of the motivation for its development.
- **Section 2: Systems Concepts** presents foundational concepts for systems and software engineering, including an overview of system components, operational context, lifecycle models, and roles.
- **Section 3: Risk Management** describes general risk management concepts, including risk management activities, approaches for analyzing risks, and options for handling risks.
- **Section 4: Security/Resilience Risk Management** provides an overview of security/resilience risk concepts, including the basic elements of security/resilience risk, types of controls, and security/resilience risk mitigation strategies.
- **Section 5: SEF Structure** illustrates and describes the SEF hierarchy of domains, goals, and practices and highlights key attributes of the SEF.

Part 2: Framework and Guidance

Part 2 presents SEF practices organized in a hierarchy of domains and goals. It also provides detailed guidance for each goal and practice. The sections in Part 2 align with the SEF's three domains:

- **Domain 1: Engineering Management** defines planning and management activities across the systems lifecycle for security/resilience engineering. Program managers and engineering leads will find this guidance useful.
- **Domain 2: Engineering Activities** describes a set of security/resilience engineering and development practices across the systems lifecycle, beginning with requirements specification and continuing through system O&S. Program managers, engineering leads, engineering and development technical personnel, and security/resilience subject matter experts (SMEs) will find this guidance useful.
- **Domain 3: Engineering Infrastructure** focuses on the engineering infrastructure. It includes practices for selecting, procuring, and integrating software, tools, and technologies that support a program's security/resilience engineering and development activities. It also includes practices for managing the engineering infrastructure to ensure that security/resilience risks are being managed appropriately. The audience for this domain is broad; program managers, engineering leads, engineering and development technical personnel, information technology (IT) managers, the IT support group, and security/resilience personnel will find this guidance useful.

Part 3: Appendices

Part 3 comprises the three appendices. Readers who want information about acronyms, terms, and references used throughout this report will find Part 3 a useful resource. Part 3 includes the following sections:

- **Appendix A: Acronyms** presents a list of the acronyms used throughout the SEF.
- **Appendix B: Glossary** defines the terms used in the SEF.
- **Appendix C: References** lists documents used as source information for the SEF.

Part 1
Introduction and Key Concepts

This part of the report sets the context for the SEF. Understanding core concepts related to systems engineering, software engineering, and risk management establishes a foundation for applying SEF practices. These foundational concepts facilitate interpreting and implementing SEF guidance, putting programs in position to design, develop, and operate software-reliant systems that are secure/resilient.

1 Introduction

Software is a growing component of modern business- and mission-critical systems. As organizations become more dependent on software-driven technology, security/resilience[1] risks to their missions also increase. Managing these risks is too often deferred until after deployment due to competing priorities, such as satisfying cost and schedule objectives. However, experience shows that failure to address these risks early in the systems lifecycle increases operational impact and mitigation costs and severely limits management options. As a result, acquisition programs should start managing a system's security/resilience risks early in the lifecycle and continue throughout the system's lifespan.

To effectively address security/resilience risks early in the lifecycle, engineers and developers must adopt the mindset of a malicious actor when assessing and managing those risks. This proactive approach enables engineers and developers to identify and remediate vulnerabilities that can lead to security/resilience risks. Cyber attacks are designed to exploit vulnerabilities in a system's software components and bypass protective controls, which makes software a focal point for early lifecycle risk analysis. Software must be architected and designed with the knowledge that it must function as intended in an increasingly contested, challenging, and interconnected cyber environment.

1.1 Software Assurance

Software assurance is 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 [CNSS 2010]. Software assurance was legislatively mandated for the Department of Defense (DoD) in the *National Defense Authorization Act for Fiscal Year 2013* [U.S.C. 2013]. The pursuit of software assurance is a worthy goal that must be translated into practical methods that acquirers, engineers, and developers can apply throughout the systems 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 that software provided to military aircraft pilots increased from 8% to 80%. At the same time, the size of software in military aircraft grew from 1,000 lines of code in

[1] The SEF is a risk-based framework that addresses two core concepts: security and resilience. With the SEF, a common set of risk management methods, tools, and techniques is used to manage both. The risks considered during an assessment and the set of controls that are available for risk mitigation are influenced by the perspective that is adopted—either security, resilience, or (in many cases) a blend of the two. Because of the related nature of security and resilience, the term *security/resilience* is used throughout the SEF. See Section 4, Security/Resilience Risk Management, for more information about managing security/resilience risks.

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

the F-4A to 1.7 million lines of code (MLOC) in the F-22. This trend is expected to continue over time [Dvorak 2009]. As software exerts more control over complex systems, like military aircraft, the potential risk posed by vulnerabilities will increase in kind.

1.2 Security Engineering Framework (SEF)

Software assurance can be decomposed into two main objectives: (1) ensure that software functions as intended and (2) produce software that is free of vulnerabilities. While both objectives are important, achieving a state entirely free of vulnerabilities is an idealistic goal that cannot be achieved in practice.

A more realistic goal is to manage the risk associated with vulnerabilities. As a result, software assurance can be interpreted as the level of confidence that (1) a software-reliant system will behave as expected and (2) its security/resilience risks will be kept within an acceptable tolerance across the systems lifecycle.

The SEF focuses on the more realistic interpretation of software assurance: managing a system's security/resilience risks across the systems lifecycle.

The SEF is a collection of software-focused engineering practices for managing security/resilience risks across the systems lifecycle, starting with requirements definition and continuing through operations and sustainment (O&S). It provides a roadmap for building security/resilience into software-reliant systems prior to deployment and maintaining the system's security/resilience capabilities during O&S. SEF practices help ensure that engineering processes, software, and tools are secure/resilient, thereby reducing the risk that attackers will disrupt program and system information and assets. Acquisition programs can use the SEF to assess their current security/resilience engineering practices and chart a course for improvement, ultimately reducing security/resilience risks in deployed software-reliant systems.[3]

1.3 SEF Problem Space

The SEF comprises a collection of leading practices for managing security/resilience risks across the systems lifecycle. When developing the SEF, a key goal was to develop an approach capable of managing the complexity of today's security/resilience risks. Many sources of complexity originate in the network of personnel, processes, and technologies that form the foundation of an organization and its operational environment. The following perspectives are useful in describing this complexity [Alberts 2014]:

- **The software perspective** focuses on building security/resilience controls into a software-reliant system, not treating security/resilience as an add-on feature that will be addressed during software sustainment activities. This perspective requires addressing

[3] The SEF documents security/resilience engineering practices across the systems lifecycle. Other areas of security and resilience practice—such as program management, supplier management, and support—are beyond the scope of the SEF. Other frameworks and models address these areas of practice.

security/resilience concerns from the earliest phases of the systems lifecycle through the operation, sustainment, and evolution of deployed systems.

- **The socio-technical perspective** stresses the prominent role of personnel in creating, using, and maintaining technologies. It also highlights the role of personnel in causing and preventing cyber attacks.
- **The cyber-physical perspective** emphasizes the notion that cyber attacks can produce consequences in the physical world. Cyber-physical systems make it possible for software to directly interact with events in the physical world.
- **The mission perspective** highlights the effect that cyber attacks have on the mission an individual, group, or organization is pursuing. As a result, security/resilience risk management must extend beyond the boundary of a technical system and consider the impact on the mission.
- **The system-of-systems (SoS) perspective** describes how a system must function as part of a multi-system environment to achieve stakeholders' objectives. This complex environment affects how security/resilience is analyzed and managed. This perspective also illustrates the complex nature of cyber attacks and how they typically include many systems that are managed by multiple, independent organizational entities.
- **The compliance perspective** describes the range of security/resilience guidelines, specifications, and laws with which an organization must comply. As a result, security/resilience risk management must consider and, when appropriate, incorporate the practices and controls specified in relevant guidelines, specifications, and laws. Each sector may be required to comply with a specific and unique set of mandated requirements. However, most mandated security/resilience requirements share a common set of principles and characteristics.

Most approaches to managing security/resilience address one or two of the above perspectives. The SEF is designed to consider all six of these perspectives.

1.4 SEF Development History

The SEF's lineage traces back to previous research that the Software Engineering Institute (SEI) conducted in the field of cybersecurity engineering. In 2010, SEI researchers identified the need for improved cybersecurity engineering early in the systems lifecycle. At that time, software security/resilience solutions primarily focused on O&S. However, there was an emerging need to develop solutions for building security/resilience into systems rather than waiting to address cyber risks during operations.

This initial research into cybersecurity engineering resulted in the *Security Engineering Risk Analysis (SERA) Method*, a scenario-based approach for analyzing complex cybersecurity risks in systems across their lifecycle and supply chain. This method, published in 2014 [Alberts 2014], provides a platform for analyzing risk in systems that are being acquired and developed.

While developing the SERA Method, SEI researchers identified a need within the cyber community to establish more effective security/resilience practices across the systems lifecycle. As a

result, SEI researchers initiated two related research activities: (1) the *Software Assurance Framework (SAF)* and (2) the *Acquisition Security Framework (ASF)*.

- **The SAF** documents cybersecurity engineering practices that programs can apply across the acquisition lifecycle. These practices can be used to assess an acquisition program's current cybersecurity engineering practices and chart a course for improvement, ultimately reducing the cybersecurity risk of deployed systems. SEI research into the SAF began in 2016, and the SEI published a prototype version in 2017 [Alberts 2017a].
- **The ASF** defines a process management approach for enabling security/resilience engineering across the systems lifecycle and supply chain. Initial research into the ASF began in 2016 and focused on defining key concepts and principles [Alberts 2017b]. SEI researchers began developing ASF practices in 2020, building on ASF foundational concepts and principles. They developed and documented leading practices for managing security/resilience across the systems lifecycle and supply chain. The ASF, initially published in 2022 [Alberts 2022], is a broad framework that includes program management, engineering, and supplier practices.

In 2022, SEI researchers identified a need to explore engineering practices in greater depth. This need ultimately led to the development of the SEF. SEI researchers used engineering practices from the SAF and ASF as the starting point for developing the SEF. They refined these practices based on additional research into leading practices for security/resilience engineering. In addition, SEI researchers developed in-depth guidance that elaborates on leading engineering practices and describes how to perform them. This report documents the results of that work.

2 Systems Concepts

Many definitions exist for the term *system*. Researchers in dependability and secure computing have defined a system as an entity that interacts with other entities (i.e., other systems), including hardware, software, humans, and the physical world. They note that the function of a system establishes what the system is intended to do and is described by the functional specification in terms of functionality and performance [Avizienis 2004].

The DoD *Systems Engineering Guidebook* views a system as "an aggregation of system elements and enabling system elements to achieve a given purpose or provide a needed capability" [DoD 2022e]. The National Institute of Standards and Technology (NIST) uses the following definition: "A system is a combination of interacting elements organized to achieve one or more stated purposes" [NIST/DOC 2020].

While these definitions have been developed for different audiences, they have a common theme: a combination of elements or components that interact to achieve a purpose or objective. This theme is reflected in the definition that is used in the SEF: A system is an arrangement of components that work together to achieve a given purpose or provide a needed capability. A system, illustrated in Figure 1, can be divided into a hierarchy of elements that includes subsystems and components.

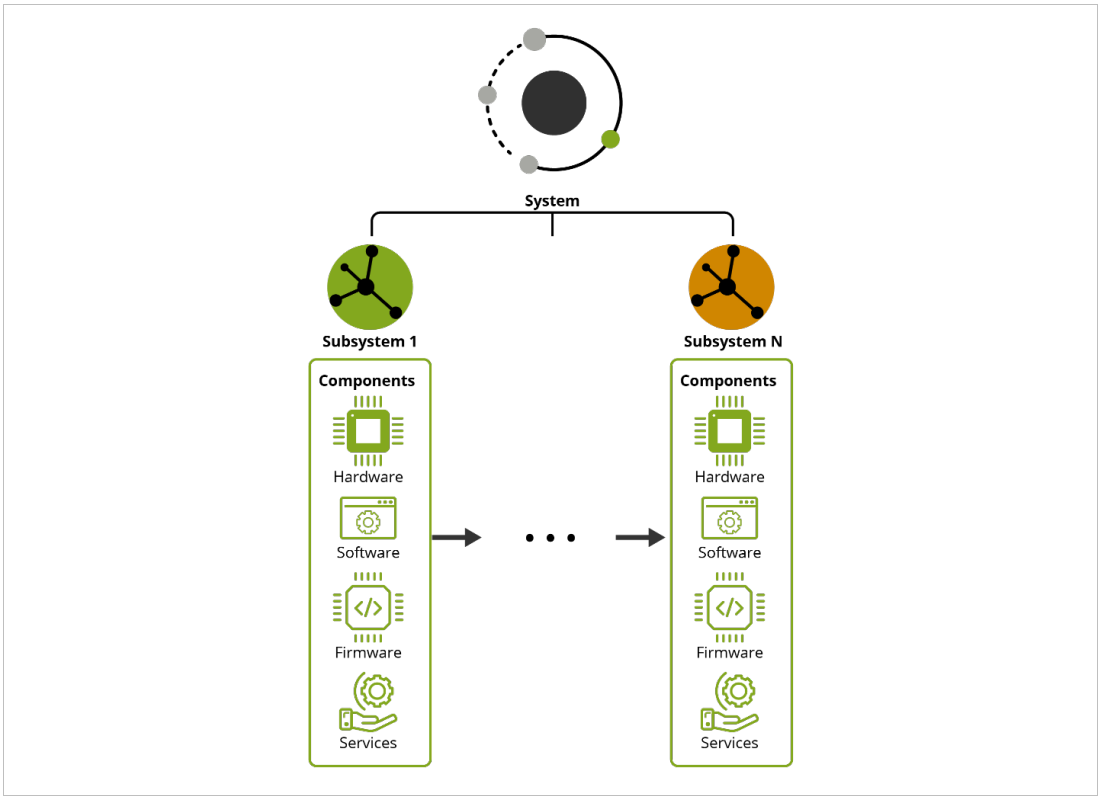

Figure 1: System Hierarchy: System, Subsystems, and Components

A complex system can be decomposed into multiple subsystems with specified interfaces among the system and subsystems. In this context, a *subsystem* is an integrated set of components that interacts with the greater system to perform a function. A subsystem must be integrated with other subsystems and components to compose a system. The designation of subsystems is often based on information flows within the system.

System components occupy the next level of the system hierarchy. A *component* is defined as one of multiple elements that compose a system or subsystem. The following system components are covered in the SEF:

- **Hardware**—the material physical components of a system
- **Software**—a set of instructions, data, or programs used to operate computers and execute specific tasks
- **Firmware**—computer programs and data stored in a hardware device's read-only memory (ROM) or programmable read-only memory (PROM) that provide instructions about how the hardware/device is designed to operate
- **Services**—infrastructure, platform, or software that is hosted or provided by a third party

A component can be either custom developed or available off the shelf. A custom component is specifically designed or adapted for use in a system or subsystem and is not generally available in the marketplace. In contrast, an off-the-shelf component is available as a stock item or commodity; it is not specially designed or custom made but can provide tailorable options. Off-the-shelf software components include commercial-off-the-shelf (COTS) software, government-off-the-shelf (GOTS) software, and open source software (OSS).

2.1 Software-Reliant Systems

A *software-reliant system* is one whose behavior (e.g., functionality, performance, safety, security, interoperability) depends on software in some significant way [Bergey 2009]. Any system where software influences the system's design, construction, deployment, and evolution is considered to be a software-reliant system. Examples include individual software applications, information systems, embedded systems, and cyber-physical systems. The SEF is designed specifically to address security/resilience in software-reliant systems with an emphasis on cyber-physical systems.

A *cyber-physical system* is an engineered system that is built from and depends on the seamless integration of computational algorithms and physical components. Cyber-physical systems merge the physical and virtual worlds, integrating objects, data, and services [NIST 2022]. Examples include industrial control systems, automobiles, robotics systems, building controls, implantable medical devices, the smart grid, and DoD weapon systems. Cyber processes monitor and collect data from physical processes, such as an automobile's steering or a hospital patient's vital signs.

Cyber-physical systems are networked, making their data globally available to other processes. Therefore, these systems make it possible for software to directly interact with events in the physical world. The cyber-physical perspective emphasizes the notion that cybersecurity attacks can produce consequences in the physical world.

2.2 Real-Time and General-Purpose Systems

A *real-time system* is one in which computation must be performed during the actual time that an external process occurs, allowing computational results to respond to those external processes [DAU 2024]. It enables real-time control over hardware resources by providing deterministic behavior and predictable response times. Because it can manage concurrent tasks, a real-time system ensures consistent operation even under extreme loads and varying conditions. Examples of real-time systems include industrial control systems, automobile-engine fuel injection systems, medical imaging systems, command-and-control systems, and weapon systems.

A *general-purpose system* is designed to be user friendly, run a variety of applications, and support multiple users and devices. As a result, it often prioritizes functionality and diversity over speed and reliability. A general-purpose system is thus more flexible and versatile than a real-time system, but it does not guarantee a specific level of performance or responsiveness. It is common for a general-purpose system to experience delays, errors, or crashes, depending on the workload and processing resources available. Examples of general-purpose systems include mainframe computers, servers, laptops, desktop computers, smartphones, and tablets.

Security/resilience risks are often managed differently in real-time and general-purpose systems. One key difference is the tradeoff among quality attributes for each type of system. Quality attributes are the functional and nonfunctional requirements that are used to evaluate system performance [Hilburn 2023]. Examples of quality attributes are performance, security/resilience, reliability, interoperability, usability, portability, maintainability, and scalability. Depending on the system being designed and developed, some quality attributes are more important than others. Quality attributes are typically prioritized differently in real-time and general-purpose systems. A key tradeoff that must be considered is performance versus security/resilience requirements.

For example, consider how the performance and security/resilience tradeoff is addressed in the two types of systems. A security/resilience requirement that introduces latency into a system's processing could introduce an unacceptable performance risk in a real-time system. In a military operation, a system's response time can be the difference between mission success and failure. As a result, an ordnance could miss its target or information might be received a few seconds too late by a command-and-control system. In both cases, system processing delays could result in the failure of an operational mission.

As illustrated in the example, engineers and developers might decide to accept a security/resilience risk to meet a real-time system's performance requirements. However, that tradeoff would likely be viewed very differently for a general-purpose system, such as an enterprise accounting system. The performance requirements of an accounting system are very different from those of a weapon system or a command-and-control system. Accounting processes can tolerate a degree of processing delays and system crashes that are unacceptable in a real-time system. The latency introduced by the security/resilience requirement likely will not affect the accounting system's performance in meaningful way.

The security/resilience practices documented in the SEF apply to both real-time and general-purpose systems. However, they will be implemented quite differently in those systems based on disparities in the systems' risks and tradeoffs.

2.3 Operational Context

A system that is being acquired and developed will ultimately be deployed in an operational environment and support one or more missions. It will also exchange data and services with other independently managed systems as part of a broader SoS environment. As a result, it is important to understand a system's intended operational environment when designing and developing a system and its components. A natural place to start is with the operational mission.

Military organizations commonly use the terms *mission* and *mission thread*:

- **Mission**—a set of objectives and goals to be achieved in a specific operational environment [DoD 2020b] (In more colloquial terms, a mission is a specific task that an individual or group is assigned to perform. An individual or group must perform several activities or steps in pursuit of a mission.)
- **Mission thread**—a sequence of end-to-end activities and events presented as a series of steps to achieve an objective or goal (e.g., a mission) [DoD 2020b] (A mission thread can be viewed as an operational model that describes how to achieve a desired outcome.)

Businesses use different terms in place of the term mission thread. For example, a *workflow* is defined as a collection of interrelated work tasks that achieves a specific result [Sharp 2008]. A workflow includes all tasks, procedures, organizations, personnel, technologies, tools, data, inputs, and outputs required to achieve the desired objectives. Business literature uses several terms synonymously with workflow, including *work process* and *business process*. The SEF treats the terms *mission thread* and *workflow* as synonyms. As a result, the phrase mission thread/workflow is used throughout the SEF.

In most operational environments, multiple networked systems support the execution of a mission thread/workflow. A *system of systems (SoS)*[4] is defined as a set or arrangement of interdependent systems that are related or connected (i.e., networked) to provide a given capability [Levine 2003].

The following characteristics are used to differentiate an SoS from a very large, complex monolithic system [Maier 1996]:

- **Managerial independence.** The management of each system within an SoS is independent from the management of the other systems.
- **Operational independence.** Each system within an SoS provides useful functionality apart from other systems.
- **Evolutionary character.** Each system within an SoS grows and changes independently of other systems over time.
- **Emergent behavior.** Certain behaviors of an SoS arise from the interactions among the individual systems and are not embodied in any of the individual systems.

[4] An *SoS* is defined as a set or arrangement of interdependent systems that are related or connected (i.e., networked) to provide a given capability [Levine 2003].

- **Geographic distribution.** Individual systems within an SoS are dispersed over large geographic areas.

An SoS provides information and services that are essential for the successful execution of a mission thread/workflow. Cyber attacks that have the potential of affecting a mission can target any system within an SoS environment, creating complex attack paths[5] that the organization must consider during a risk analysis.

2.4 Lifecycle Approach

The SEF uses the generic term *systems lifecycle* to describe the activities involved in acquiring and developing a software-reliant system—from initial concept through system disposal. A number of specific lifecycle models can be implemented, such as Waterfall, Agile, DevOps, and DevSecOps:[6]

- **Waterfall.** A Waterfall lifecycle is a sequential model for system development. It is divided into phases, where the output of one phase becomes the input of the next phase. A given phase in the lifecycle must be complete before the next phase starts. The phases in a Waterfall lifecycle do not overlap.
- **Agile.** Agile development is not a sequential model. It uses an iterative approach to system design and development. When applying Agile, teams allocate lengthy requirements, build, and test phases into smaller work increments, allowing them to deliver software more frequently. Agile provides a more flexible approach than the Waterfall lifecycle's strictly defined phases. In software development, many programs are moving away from the Waterfall lifecycle and are applying Agile's iterative and incremental approach.
- **DevOps and DevSecOps.** Other software and systems development approaches, such as DevOps and DevSecOps, accelerate delivery through automation, collaboration, fast feedback, and iterative improvement. Drawing on the Agile approach, DevOps and DevSecOps expand on the cross-functional approach of building and shipping software. In the DevOps approach, development (Dev) and operations (Ops) teams work collaboratively across the lifecycle to increase the speed and quality of deployed software. DevSecOps adds security (Sec) personnel to the collaborative team.

While each lifecycle implements a unique model for developing software and systems, all address a common set of engineering activities, including the following:

- requirements
- architecture
- implementation
- test and evaluation (T&E)

---

[5] An *attack path* is a pathway or method that an attacker uses to access a system and exploit the system's weaknesses and vulnerabilities.

[6] DevSecOps stands for Development, Security, and Operations.

- deployment
- operations and sustainment (O&S)

The SEF is designed to expand the focus of systems lifecycle models to include security/resilience. The SEF defines goals, practices, and guidance for integrating security/resilience into existing systems and software engineering activities across the lifecycle, regardless of the lifecycle model used.

2.5 Integrated Systems and Software Engineering Practices

Software is fundamental to the performance of today's engineered systems [SEBoK 2023]. As software has grown to represent much of a system's functionality, the line between systems engineering and software engineering is no longer clear cut. Software is not just a separable part of a system; it is an integral part of the capabilities that a system provides. In fact, software shapes a system's requirements, architecture, and design; contributes to much of a system's complexity; and drives much of a system's cost and schedule during development.

The systems lifecycle must account for software's prominent role in modern systems by integrating software activities with system-focused activities. Software requirements and architecture practices must align with system requirements and architecture practices. Similarly, software integration and test practices should align with system integration and test practices [SEBoK 2023].

System and software development/implementation activities (e.g., coding, code analysis, hardware fabrication) might be performed separately. However, individual components (i.e., hardware, software, firmware, services) are assembled into one system during integration activities. SEF practices reflect this integrated view of systems and software engineering practices across the lifecycle.

2.6 Organizational Paradigm

The SEF is based on an organizational paradigm that includes the following entities:

- **Acquisition program.** An *acquisition program* is an organizational unit responsible for acquiring and developing a software-reliant system in response to a defined need. It is the focal point of the SEF. Acquisition program personnel oversee all engineering and development aspects of the system. A program's technical personnel can also perform some engineering activities, such as system requirements specification and system architecture definition and analysis.
- **Contractor.** Acquisition program personnel are responsible for managing development activities, which contractors typically perform. In this context, a *contractor* is an organization that enters into an agreement (i.e., contract) with an acquisition program to provide custom components to the program. Contractors are responsible for many engineering and development activities, including requirements specification, architecture development, and implementation and coding activities. They also can perform management/oversight roles, including serving as the prime contractor and assuming the role of system integrator. The *prime*

contractor is responsible for managing system development, including organizing and managing subcontractors. The *system integrator* assembles components and subsystems and ensures that the system functions as a whole. In many instances, a single organization assumes both prime contractor and system integrator roles.

- **Supplier.** Numerous suppliers also support an acquisition program. A *supplier* is an organization that provides off-the-shelf components to the program and its contractors.[7] The off-the-shelf components are then integrated into the system that is being acquired and developed. Off-the-shelf components include COTS software, GOTS software, OSS, and cloud services.
- **Acquisition enterprise.** An enterprise is an organization with a defined mission and boundary. It uses information systems to execute its mission and is responsible for managing its own risks and performance. An *acquisition enterprise* is the broader organization in which the acquisition program resides. It typically provides services that support program execution, such as operational test and evaluation (OT&E), authorization to operate (ATO), O&S, and independent assessments. Some acquisition enterprises also provide security/resilience subject matter experts (SMEs) to assist program personnel.
- **Third-party assessor.** As a system is being designed and developed, the program manager might decide to obtain an impartial perspective on the status of the acquisition program or system by chartering an independent assessment. Independent assessments comprise a variety of assessment types that are performed by personnel who are not connected with a program or system. Examples of independent assessments include programmatic assessments, technical assessments, compliance assessments, process assessments, blue team assessments, and red team assessments. These assessments provide an objective, unbiased review of a program or system. An independent assessment team can come from an organizational unit within the acquisition enterprise that is not connected with the acquisition program. However, in many instances, a program manager or stakeholder from the acquisition enterprise will contract with a third-party assessor to perform an independent assessment. A *third-party assessor* is an assessment team from an organization without connections to or relationships with the acquisition program, its parent enterprise, or its contractors.
- **End-user organization.** After a system is deployed into its operational environment and enters the O&S phase of the systems lifecycle, it is actively used and supported. In system development, the end-user organization includes the personnel who ultimately use or are intended to use the system. The *end-user organization* stands in contrast to the O&S organization, which is responsible for operating and maintaining deployed systems.
- **O&S organization.** The *O&S organization* manages system risks (including security/resilience risks) during operations while cost-effectively ensuring that the system supports its missions well. It manages modifications, upgrades, and future increments of the system; manages changes to system support artifacts and activities; and implements process

[7] The SEF differentiates between contractors and suppliers. A contractor develops custom components under a formal agreement with the program. Custom components are developed to meet a specific set of requirements. In contrast, an acquisition program procures or licenses off-the-shelf components from suppliers. Off-the-shelf components are stock items or commodities.

improvements where appropriate. The O&S organization is responsible for the decommissioning and disposal of a system when it reaches its end of life.

The SEF was developed from the perspective that the acquisition program, O&S organization, and end-user organization are separate entities and have independent management chains.

2.7 Roles

SEF practices describe activities that must be performed when designing and developing a system and system components. *Practices* are defined to be independent of roles; however, for clarity, SEF guidance sometimes refers to roles that are typically involved in performing those practices. The following roles are mentioned throughout SEF guidance (in Part 2):

- **Program manager**—the individual with the responsibility and authority for accomplishing program objectives for the development, production, and sustainment of systems to meet the user's operational needs
- **Engineering lead for the program**—the individual who provides technical expertise and strategic leadership for the design and development of complex software-reliant systems, including specifying requirements, defining architectures, and performing verification and validation activities
- **Engineer**—the individual who develops and manages user, system, and software requirements; leads system architecture development; evaluates design tradeoffs; or oversees verification and validation activities
- **Engineering team**—the collection of engineers developing a system
- **Architect**—the individual who defines the architecture for a software-reliant system to meet specified requirements
- **Developer**—the individual who designs and writes software code, builds software components, or tests software performance
- **Tester**—the individual who plans, prepares, and executes tests of a system to validate whether it meets its requirements and verify that it fulfills its intended purpose
- **Assessor**—the individual who conducts an assessment of a program or system to evaluate its performance
- **Assessment team**—the collection of assessors who collaborate to evaluate a program or system
- **Operator/maintainer**—the individual who monitors, supports, troubleshoots, and maintains system components to ensure that the system is functioning and meeting the needs of end users
- **O&S manager**—the individual who oversees the operation and maintenance of a deployed system
- **Information technology (IT) support group**—the collection of personnel who manage the IT systems and networks that support engineering and development activities (i.e., the engineering infrastructure), including setting up servers, configuring networks, managing databases, and monitoring system performance

- **End user**—the individual who ultimately uses or is intended to use the system
- **Security/resilience SME**—the individual who has knowledge and expertise in security/resilience technical, management, and compliance activities
- **Program/system stakeholder**—the individual, group, or organization with a vested interest (i.e., stake) in the decision making and activities of the acquisition program or the system that is being acquired, developed, and operated

These roles are mentioned throughout the SEF to provide context about the personnel who are typically tasked with applying SEF practices. However, the names of the roles are not used universally. Individual organizations use their own labels for these roles, so readers may need to translate SEF roles to their organization's unique terminology.

The systems and organizational concepts presented in this section provide important context needed to understand and apply SEF practices and guidance. Readers must also understand basic security/resilience and risk management concepts that the SEF is built on. The next two sections provide an overview of risk management and security/resilience from the SEF perspective.

3 Risk Management

The term *risk* is used universally, but different audiences attach different meanings to it [Kloman 1990]. In fact, the details about risk and how it supports decision making depend on the context where it is applied [Charette 1990]. For example, safety professionals view risk management in terms of reducing the number of accidents and injuries. A hospital administrator views risk management as part of the organization's quality assurance program, while the insurance industry relies on risk management techniques when setting insurance rates. Therefore, each industry uses a definition that is tailored to its context.

No universally accepted definition of risk exists. While specific definitions of risk might vary, a few characteristics are common to all definitions. For risk to exist in any circumstance, the following three conditions must be satisfied [Charette 1990]:

- The potential for loss must exist.
- Uncertainty with respect to the eventual outcome must be present.[8]
- Some choice or decision is required to deal with the uncertainty and potential for loss.

These three characteristics can be used to forge a basic definition of risk. Most definitions focus on the first two conditions—loss and uncertainty—because they are the two measurable aspects of risk. Thus, the essence of risk, no matter what the domain, can be succinctly captured by the definition: Risk is the probability of suffering harm or loss [Dorofee 1996].

3.1 Risk Management Activities

Risk management is a systematic approach for minimizing exposure to potential losses. It provides a disciplined environment for continuously assessing what could go wrong (i.e., assessing risks), determining which risks to address (i.e., setting mitigation priorities), and implementing actions to address high-priority risks and bring those risks within tolerance. Risk management activities can be grouped into two main phases: assess and manage, both of which are discussed in this section.

Assess

The first phase, *assess*, is the process of identifying, evaluating, and prioritizing risks. It includes two activities:

- **Identify**—transforming concerns and uncertainties into distinct, tangible risks that are documented in a prescribed format (The objective is to anticipate what could go wrong.)

[8] Some researchers separate the concepts of certainty (the absence of doubt), risk (where the probabilities of alternative outcomes are known), and uncertainty (where the probabilities of possible outcomes are unknown). However, because uncertainty is a fundamental attribute of risk, this report does not differentiate between decision making under risk and decision making under uncertainty.

- **Analyze**—evaluating probability, impact, and risk exposure for each risk and using those measures to establish risk mitigation priorities (The objective is to gain a better understanding of risks by examining risk-related data in relation to a set of evaluation criteria.)

The assess phase provides stakeholders with a broad perspective of what could go wrong, providing them with an opportunity to take action to address high-priority risks before they become problems.

Manage

The second phase, *manage*, is the process of handling and tracking risks over time. It comprises two activities:

- **Handle/treat**—developing and implementing a plan to address or handle each risk based on available mitigation options (e.g., accept, avoid, transfer, watch, mitigate) (The objective is to take proactive action to address high-priority risks before they become problems.)
- **Track**—monitoring risks and mitigation plans over time (The objectives are to identify and manage (1) the effectiveness of mitigation plans, (2) changes that can affect a risk's measures, and (3) conditions that indicate new risks may have emerged.)

In the context of risk management activities, there is no universal standard. Risk terminology varies across disciplines. Risk management methods, tools, and analytical approaches also vary. Readers may need to translate the risk management concepts embedded in the SEF to their organization's terminology, processes, and methods. While specific terminology and implementations vary, the core risk management principles and concepts apply across disciplines. Risk management provides a foundation for minimizing exposure to potential losses and putting organizations and programs in a position to succeed.

3.2 Documenting Risks

A key aspect of risk identification is documenting risk information, which is typically done in a *risk statement*. Creating a risk statement is an important aspect of identifying, managing, and communicating risk information. It provides a succinct and unique description of each risk and communicates the potential adverse event and its consequences if the risk is realized. A clear and concise risk statement ensures that personnel across a program develop a common understanding of the potential problem.

An if-then format is often used to form risk statements. The *if* part of the statement conveys the potential adverse event, while the *then* portion expresses the resulting consequences [DoD 2017]. Another approach to documenting a risk is a condition-consequence format. The *condition* part of the statement establishes the circumstances that are causing concern, while the *consequence* portion expresses the resulting consequences [Dorofee 1996]. In general, the disciplined use of a structured format helps in describing and communicating risks to program stakeholders.

3.3 Risk Analysis

Risk analysis is an activity for evaluating and prioritizing risks. In general, three measures are associated with any risk:[9]

- **Probability**—a measure of the likelihood that the risk will occur
- **Impact**—a measure of the loss that occurs when a risk is realized
- **Risk exposure**—a measure of the magnitude of a risk based on current values of probability and impact

Risk evaluation establishes probability, impact, and risk exposure measures for each identified risk. Risk evaluation can be qualitative or quantitative:

- **Qualitative risk evaluation.** This type of risk evaluation is an approach based on the assignment of descriptors such as low, medium, and high to evaluate risk [NIST 2021]. Assessors define a set of evaluation criteria when they are preparing to conduct a risk assessment. These criteria are typically documented in the risk management plan, and they include a set of qualitative measures for assessing probability and impact. A risk exposure matrix provides a way of estimating the magnitude of a risk based on the current values of probability and impact.
- **Quantitative risk evaluation.** This type of risk evaluation assigns numerical values to impact and probability based on statistical probabilities and valuation of loss or gain [NIST 2021]. A quantitative risk evaluation provides more objective data than a qualitative risk analysis because it is based on measurable risk data. A common problem with a quantitative risk evaluation is insufficient data. In practice, most organizations implement a qualitative risk evaluation approach.

Risk prioritization is the process of determining the order in which risks should be addressed, based on probability, impact, and risk exposure measures. Assessors must define decision-making criteria for prioritizing risks and include those criteria in the organization's risk management plan.

3.4 Options for Handling Risks

Risk handling (i.e., risk treatment) is where plans are developed and implemented based on available options. Risk management methods offer options for handling a risk [DoD 2017], including the following:

- **Accept.** If a risk occurs, its consequences will be tolerated; no proactive action to address the risk will be taken. When a risk is accepted, the rationale for doing so is documented.
- **Avoid.** Activities are restructured to eliminate the possibility of a risk occurring.
- **Transfer.** A risk is shifted to another party (e.g., through insurance or outsourcing).

[9] A fourth measure, time frame, is sometimes used to measure the amount of time that elapses before a risk is realized or the amount of time during which action can be taken to prevent a risk.

- **Watch.** A risk is monitored for changes to its measures (i.e., impact, probability, risk exposure).
- **Mitigate.** Action is taken to reduce or contain a risk. This can include implementing controls/countermeasures to address a risk.

Options for handling risks are selected based on a risk's measures (i.e., probability, impact, and risk exposure) and relative priority, which are established during risk analysis. High-priority security/resilience risks are generally mitigated.

4 Security/Resilience Risk Management

At its core, the SEF is a risk-based framework that addresses two core concepts: security and resilience:

- **Security** establishes and maintains protective measures that enable an organization to perform its mission or critical functions despite risks posed by threats to its systems. Protective measures may involve combining protection, detection, response, and recovery to provide the basis for an organization's risk management approach [NIST 2021].
- **Resilience** is the ability to prepare for and adapt to changing conditions and recover rapidly from disruption. Resilience includes the ability to withstand and recover from deliberate attacks, accidents, or naturally occurring threats or incidents [NIST/DOC 2020].

Risk management provides the foundation for managing security and resilience. In fact, risk management methods, tools, and techniques are used to manage both. However, security and resilience view risk from different perspectives: Security considers risks from a *protection* point of view, whereas resilience considers risk from a perspective of *adapting* to conditions, stresses, attacks, and compromises. As shown in Figure 2, there is some overlap between the risk perspectives of security and resilience. At the same time, security and resilience each have unique risks and mitigations.

Figure 2: Risk Perspectives: Security Versus Resilience

The SEF uses the term *security/resilience* because of the related nature of security and resilience and the overlap in their risk profiles. Ultimately, the same set of practices can be used to manage security and resilience risks. The risks considered by a risk assessment and the set of controls that are available for risk mitigation are influenced by the perspective that is adopted—either security, resilience, or, in many cases, a blend of the two.

4.1 Security/Resilience Risk Management Planning

Risk management is a systematic approach for minimizing exposure to potential losses. From a security/resilience perspective, risk management comprises the activities for managing security/resilience risks to organizational operations, organizational assets, individuals, other organizations. Activities for managing security/resilience risks include (1) establishing the context for risk-related activities, (2) assessing risks, (3) managing risks, and (4) monitoring risks over time. [10]

Over the years, many risk assessment methods, tools, and techniques have been developed. Program managers and engineers need to select a suite of methods, tools, and techniques that meet their needs and requirements. As directed in *DoD Instruction 8510.01, Risk Management Framework for DoD Systems*, defense programs are required to follow the National Institute of Standards and Technology (NIST) Risk Management Framework (RMF) as defined in NIST Special Publication 800-37, *Risk Management Framework for Information Systems and Organizations*, to inform the acquisition processes for all DoD systems [DoD 2022a, NIST/DOC 2018]. The RMF provides a comprehensive, flexible, repeatable, and measurable seven-step process that organizations can use to manage security/resilience risks.[11] This process is designed to improve the security/resilience characteristics of a system, strengthen an organization's risk management processes, and encourage reciprocity among organizations. This process also promotes developing security and privacy capabilities into systems throughout the lifecycle.

No matter which method a program chooses to implement, security/resilience risk management begins by developing a plan for assessing and managing security/resilience risks. A key aspect of planning for security/resilience risk management is selecting the risk model and analytic approach that are implemented in the risk assessment process. The *risk model* establishes the risk factors to be assessed and the relationships among the factors [NIST/DOC 2012]. Common risk factors included in risk assessments are threat, vulnerability, impact, likelihood, and predisposing conditions. The *analytic approach* defines the assessment approach (i.e., quantitative, qualitative, semi-quantitative) and the analysis approach (i.e., threat oriented, asset/impact oriented, vulnerability oriented) [NIST/DOC 2012]. The remainder of this section examines key concepts for assessing and managing security/resilience risks.

4.2 Security/Resilience Controls

Security/resilience controls are the safeguards or countermeasures prescribed for an information system or organization to protect the confidentiality, integrity, and availability (CIA) of the

[10] The definition of security/resilience risk management and the risk management activities are adapted from NIST SP 800-30 [NIST/DOC 2012].

[11] The RMF steps are Prepare, Categorize, Select, Implement, Assess, Authorize, and Monitor. The Categorize, Select, and Implement steps are performed during system acquisition and development. The Assess and Authorize steps are performed during a system's ATO activities. The Monitor step is performed during O&S. Refer to the report *Risk Management Framework for Information Systems and Organizations* for detailed information about the RMF and its seven-step process [NIST/DOC 2018].

system and its information [NIST/DOC 2020]. There are three basic types of security/resilience controls:

- **Technical controls**—safeguards or countermeasures that are primarily implemented and executed through mechanisms contained in system components [NIST/DOC 2012] (Examples of technical controls include firewalls, intrusion detection systems [IDSs], encryption, and identification and authentication mechanisms.)
- **Physical controls**—mechanisms that deny unauthorized access to facilities, equipment, and resources, and protect personnel and property from damage or harm (Examples of physical controls are card readers, cameras, motion sensors, intruder alarms, equipment inventories, surge protectors, and fire protection.)
- **Administrative controls**—policies, procedures, or guidelines that define personnel and organizational practices in accordance with the organization's security goals (Examples of administrative controls are security education training and awareness programs, password management policies, and incident response planning.)

From a systems lifecycle perspective, all three types of security/resilience controls must be implemented. During system design and development, the emphasis is on implementing technical controls at the system level. These controls are called *system controls*. However, engineers must consider the broader operations and SoS environments when specifying controls. The system being designed and developed will inherit security/resilience controls from one or more systems across an enterprise. These controls are called *common controls*. From a risk management perspective, engineers must specify both system controls and common controls. The controls that are initially selected during design and development provide a foundation for assessing and managing the system's security/resilience risks across the lifecycle.

4.3 Security/Resilience Risk Concepts

The Committee on National Security Systems (CNSS) defines risk as "a measure of the extent to which an entity is threatened by a potential circumstance or event, and typically is a function of: (i) the adverse impact, or magnitude of harm, that would arise if the circumstance or event occurs; and (ii) the likelihood of occurrence" [CNSS 2022]. In general terms, security/resilience risk is a measure of (1) the likelihood that a threat will exploit a vulnerability to produce an adverse consequence or loss and (2) the magnitude of the loss [Alberts 2014]. The following are the core components of security/resilience risk:

- **Threat**—a cyber-based act, occurrence, or event with the potential to harm an information system through unauthorized access, destruction, disclosure, or modification of data, and/or denial of service [NIST/DOC 2020]
- **Vulnerability**—a defect in a system, system security procedures, architecture, security/resilience controls, or code that a threat could exploit to produce an adverse consequence or loss
- **Consequence**—the loss that results when a threat actor exploits one or more vulnerabilities

From a security/resilience perspective, a vulnerability is the passive element of risk. It exposes cyber technologies (e.g., software application, software-reliant system) to the threats and losses

those threats can produce. By itself, however, a vulnerability will not cause a cyber technology to suffer a loss or experience an adverse consequence; rather, the vulnerability makes it susceptible to the effects of a threat actor [Alberts 2002].

These three core components provide a simplified view of security/resilience risk, where a single threat actor exploits a single vulnerability in a single system to cause an adverse consequence. Many risk assessment methods are based on this simplified view of risk. However, in reality, one or more actors can exploit multiple vulnerabilities in multiple systems as part of a complex chain of events to produce a range of adverse consequences. Risk management methods must address this inherent complexity of security/resilience risks.

4.4 Identifying Security/Resilience Risks

To capture the complexity of a security/resilience risk, assessors must adopt the mindset of a malicious actor. From the actor's perspective, a security/resilience risk is a sequence of events that will degrade mission execution or cause outright mission failure. Using the actor's perspective, a security/resilience risk can be decomposed into three core events:

- **Attack path.** A malicious actor (or actors) exploits one or more vulnerabilities to traverse an SoS environment and gain access to the system that is the target of the cyber attack.
- **Cyber attack.** The actor launches a cyber attack on the targeted system by exploiting one or more vulnerabilities. The attack produces a direct consequence that targets system data (e.g., unauthorized information disclosure or theft, modification or manipulation of information or services, destruction of information, interruption of access to information or services).
- **Mission consequences.** A cyber attack's direct consequences often propagate to other systems within the SoS environment and to the mission threads/workflows supported by the system that is the target of the attack. The impacts on other systems and mission threads/workflows are indirect consequences of the attack.

Figure 3 illustrates the relationship between the three events and the general concepts of threat, vulnerability, and consequence.

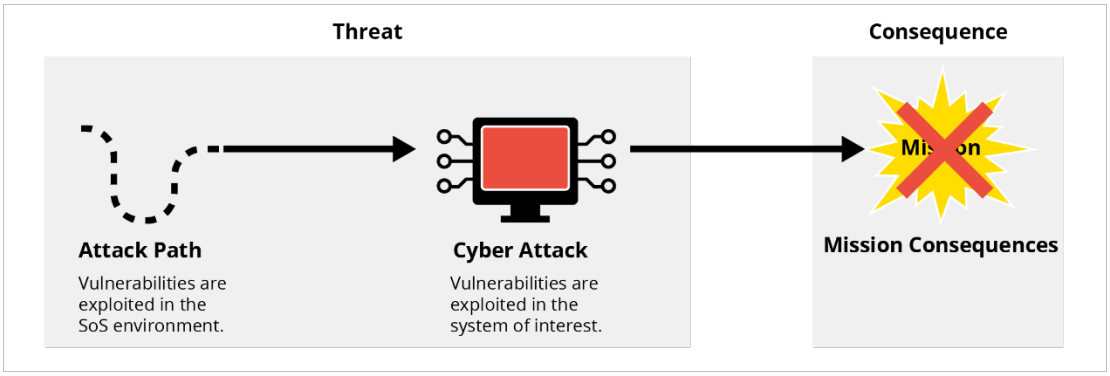

Figure 3: Mapping of Security/Resilience Events to Concepts

From a security/resilience perspective, risk is viewed as a measure of the (1) likelihood that a threat will produce a direct consequence on system data, and (2) the magnitude of the mission consequences. There are several options for handling or treating security/resilience risks (accept, avoid, transfer, watch, mitigate). High-priority security/resilience risks are generally mitigated. When developing mitigation strategies for security/resilience risks, controls that will reduce or contain those risks are typically selected.

4.5 Security/Resilience Risk Mitigation Strategies

Mitigation plans for security/resilience risks incorporate the following basic strategies:[12]

- **Protect.** Reduce vulnerability to threats and minimize any consequences that might occur.
- **Detect.** Identify the occurrence of a security/resilience threat (i.e., cyber attack).
- **Respond.** Take action to counteract a detected threat (i.e., cyber attack) and minimize consequences, losses, and damages.
- **Recover.** Restore access to and functionality of a system (or systems) after a risk's consequences, losses, and damages are realized.
- **Adapt.** Enable a sustained capability to accommodate changes in a system's risk environment, including changes to threats, vulnerabilities, mission, and technologies.

The system's risk mitigation plan specifies a set of security/resilience controls. Once a mitigation plan is developed, documented, and approved, the engineering team must obtain resources and then implement the plan.

4.6 Vulnerability vs. Weakness

A *vulnerability* is one of the core components of security/resilience risk. The CNSS defines a vulnerability as "a known weakness in a system, system security procedures, internal controls, or implementation by which an actor or event may intentionally exploit or accidentally trigger the weakness to access, modify, or disrupt normal operations of a system—resulting in a security incident or a violation of the system's security policy" [CNSS 2022]. NIST defines vulnerability as "a weakness in an information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source" [NIST/DOC 2020].

In its most basic definition, a vulnerability is a system defect that has known exploits. In the context of the SEF, an exploit is software or a sequence of commands that takes advantage of a

[12] The mitigation strategies documented in the SEF are a composite of strategies defined for security, resilience, and survivability. Cybersecurity mitigation strategies are documented in *Framework for Improving Critical Infrastructure Cybersecurity, Version 1.1* [NIST 2018]. Strategies for mitigating resilience risks for a system are presented in *Developing Cyber-Resilient Systems: A Systems Security Engineering Approach* [NIST 2021]. Finally, strategies that contribute to the survivability of a system's capabilities are addressed in the *Cyber Survivability Endorsement (CSE) Implementation Guide, Version 3.0* [DoD 2022b].

vulnerability to cause unintended or unanticipated behavior to occur in a system or its components. Therefore, a vulnerability has known attacks or exploits that can lead to risk.

Weakness is a concept that is related to vulnerability. The CNSS defines a weakness as "an attribute or characteristic that may, under known or unknown conditions, render an entity, asset, system, network, or geographic area open to exploitation or susceptible to a given hazard" [CNSS 2022]. NIST refers to a weakness as "a defect or characteristic that may lead to undesirable behavior" [NIST 2022].

In general terms, a weakness is a defect or flaw in a system, maintenance procedure, internal control, architecture, design, or implementation that has the potential to be exploited by a threat actor. In contrast to a vulnerability, a weakness, by definition, has only the potential for exploitation (i.e., no known exploits).

In the systems lifecycle, the concepts of weakness and vulnerability are important during system design activities (e.g., requirements specification, architecture development) and system implementation (e.g., coding). A *design weakness* is a defect or flaw in a system's architecture or detailed design with the potential for exploitation when implemented. For example, the requirements document for a system might specify a weak authentication protocol (e.g., single-factor authentication), or it might not require a communication channel to be encrypted. These missing or inadequate requirements are defects or flaws that provide the potential for exploitation after the system is built. From an architecture perspective, a system might have a large attack surface[13] with many points of electronic access. This is a design weakness because, when implemented, the architecture will provide threat actors with many potential attack vectors.

From an implementation perspective, a weakness is a condition in a software, firmware, hardware, or service component that, under certain circumstances, could contribute to the introduction of vulnerabilities [MITRE 2023]. A vulnerability is a defect in a software, firmware, hardware, or service component resulting from a weakness that can be exploited to produce a negative impact on the CIA of system components and their associated data [MITRE 2023]. Therefore, implementation weaknesses are errors or defects that can lead to vulnerabilities.

A weakness in a specific software product with known exploits is considered to be a vulnerability. Examples of software weaknesses include buffer overflows, structure and validity problems, channel and path errors, authentication errors, resource management errors, insufficient verification of data, and code evaluation and injection.

4.7 Managing Design Weaknesses

A key objective of the SEF is to address design weaknesses during the requirements and architecture phases of the systems lifecycle. Identifying and correcting design weaknesses as soon as possible is especially important because these weaknesses are not corrected easily after a system has

[13] A system's *attack surface* is the set of points on the boundary of a system, a system component, or an environment where an attacker can try to enter, cause an effect on, or extract data from that system, system component, or environment [NIST 2020].

been deployed. For example, an O&S organization cannot normally issue a patch to correct a fundamental security issue related to the requirements or architecture. Remediation of design weaknesses normally requires extensive changes to the system, which is costly and often proves to be impractical. As a result, systems with design weaknesses are often allowed to operate under a high degree of residual security risk, putting their associated operational missions in jeopardy. Effective security/resilience practices during the requirements and architecture phases provide a solid foundation for managing risk.

4.7.1 Requirements

Defining requirements is a critical part of developing software-reliant systems. It begins early in the lifecycle, and requirements must be managed continuously throughout a system's acquisition and development. A requirements specification records the necessary attributes, capabilities, characteristics, and qualities of the system and its software that benefit stakeholders. Multiple types of requirements are developed across the systems lifecycle, ranging from high-level and concept-focused requirements to highly technical ones. The SEF focuses on specifying, analyzing, and managing security/resilience requirements for both the system and its software components.

Engineers need to address security/resilience requirements from two key perspectives: risk mitigation and compliance. Effective risk management requires engineers to analyze the security/resilience risks to the system and its software components and establish requirements that appropriately mitigate those risks. When addressing compliance, engineers derive security/resilience requirements from applicable policies, standards, laws, and regulations.

4.7.2 Architecture

During the architecture phase of the systems lifecycle, candidate solutions are evaluated based on the system's performance requirements. The architecture phase is iterative. The goal is to select a solution that (1) achieves a balance across cost, schedule, performance, and risk objectives and (2) meets stakeholders' needs. Key outputs of the architecture phase are the system and software architectures.

A *system architecture* documents the structure of system components, their relationships, and the principles and guidelines governing their design and evolution over time.[14] The development of the system architecture should adhere to sound systems engineering practices and conform to industry standards as applicable [DoD 2022e]. It is common for a system architecture to include a functional architecture and a physical architecture. The *functional architecture* defines the system's functions and how they interact with each other to achieve the system's mission. It provides the foundation for the system architecture through the allocation of functions and subfunctions to system components, facilities, and processes [DoD 2022e].

According to the DoD's *Systems Engineering Guidebook,* the *physical architecture* consists of "one or more product structures, or views, of the physical solution" [DoD 2022e]. It illustrates

[14] This definition for *system architecture* is derived from the general definition for architecture that is documented in [DAU 2024]: An *architecture* documents the structure of components, their relationships, and the principles and guidelines governing their design and evolution over time.

how the system's physical elements and interfaces are arranged. The physical architecture documents the design for the system consistent with the functional architecture and system requirements. Development of the physical architecture is complete when the system has been decomposed into its core system components [DoD 2022e].

The architecture phase does not end with the development of the physical architecture. Design activities continue with the system's components. From a security/resilience perspective, the architectures for the system's software components are of particular interest. A *software architecture* is the set of structures needed to reason about the system, which comprises software elements, relations among them, and properties of both [Clements 2010]. It documents the high-level design of a software component. The software architecture represents the design decisions related to the overall structure and behavior of a system's software components.

Both system and software architectures help stakeholders understand and analyze how a system will achieve essential quality attributes, including performance and security/resilience. A key practice for analyzing the security/resilience of the system and software architectures is performing an architecture risk assessment.

4.7.3 Architecture Risk Assessment

During the architecture phase of the lifecycle, architects and engineers select an initial set of security/resilience controls to incorporate into the system. This initial set is then tailored and adjusted based on an assessment of security/resilience risks. A security/resilience risk assessment of an architecture (also referred to as threat modeling[15]) identifies threats and weaknesses (i.e., defects or flaws) and evaluates the risks they pose to the mission. The assessment also examines security/resilience risks resulting from the broader networked environment (i.e., inherited risks from external systems and services).

A tradeoff analysis is performed to prioritize architectural decisions across the quality attributes. Tradeoffs are evaluated among candidate design options, documented clearly, and tracked as part of a program's risk management activities. The impact of each design option on the system's quality attributes is evaluated, and the option that best addresses the system's requirements overall is selected.

Architecture risk analysis/threat modeling should be performed on both the system and software architectures. Many programs analyze only the system architecture for security/resilience risks. This approach is problematic because some design weaknesses originate in the lower level design aspects of the software architecture. Weaknesses in the software architecture may not be apparent/visible when performing a system-level architecture analysis. As a result, attackers may be able to exploit design weaknesses in the software architecture if those weaknesses are not corrected.

[15] According to NIST, threat modeling is "a form of risk assessment that models aspects of the attack and defense sides of a logical entity, such as a piece of data, an application, a host, a system, or an environment" [NIST/DOC 2020]. Threat modeling is used to identify use cases, threat agents, attack vectors and patterns, design patterns, and compensating controls needed to mitigate risk.

Risk assessments of the system and software architectures enable an engineering team to manage security/resilience risks as the system and its software components are being developed. It is part of a broader risk management strategy that requires managing security/resilience risks at both the project and product levels across the lifecycle.

4.8 Project and Product Risk Management

Acquisition programs manage engineering activities as a project. The goal of an engineering project is to design, develop, and deliver a product. From an SEF perspective, the product is a software-reliant system. Some engineering projects focus on developing and deploying new systems, while others involve providing new capabilities or upgrades to existing systems. The life of a typical system begins with an initial concept and continues through decommissioning and disposal. Over this lifespan, many engineering projects will be initiated and performed. Each engineering project must be planned and managed to address requirements and meet stakeholder objectives.

The engineering lead is responsible for managing engineering resources, including staffing, budgeting, and scheduling. An engineering project plan defines the engineering activities that will be performed across the lifecycle to develop and deploy the system. The program manager and engineering lead are responsible for acquiring the resources (e.g., funding, staffing, and tools) needed to carry out the engineering project plan. The engineering lead tracks progress against the plan, adjusts it as needed, and reports status to the program manager.

An engineering team must identify and manage project risks that affect the performance of its security/resilience engineering activities. Insufficient resources (e.g., funding, staffing, tools), schedule issues, and technical issues are examples of conditions that can affect the team's ability to conduct security/resilience engineering activities adequately; these conditions can lead to project risks. From a security/resilience perspective, the engineering team must identify and manage project risks that affect its ability to meet its security/resilience objectives.

The engineering team must also manage risks to the system that it is developing. Product risk management addresses a range of risks, including performance, reliability, safety, and security/resilience, among others. From a security/resilience perspective, the engineering team must identify and manage security/resilience risks that can affect the system during its use.

The SEF differentiates project risk management from product risk management:

- **Project risk management** is the process of identifying and managing project-level security/resilience risks, including risks related to resources, cost, and schedule.
- **Product risk management**, in contrast, is the process of managing security/resilience risks in the system that is being designed and developed and focusing on risks resulting from the exploitation of vulnerabilities in related systems and their components.

5 SEF Structure

The SEF hierarchy of domains, goals, and practices is described below and illustrated in Figure 4.

- **Domains** occupy the top level of the SEF hierarchy. A domain captures a unique management or technical perspective of managing security/resilience risks across the systems lifecycle. Each domain is supported by two or more goals, which form the next level of the SEF hierarchy.
- **Goals** define the capabilities that a program leverages to build security/resilience into a software-reliant system. Related goals are assigned to the same SEF domain.
- **Practices** inhabit the final and most detailed level in the hierarchy. Practices describe actions that support the achievement of SEF goals. In the SEF, practices are phrased as questions.[16] Related practices are assigned to the same SEF goal.

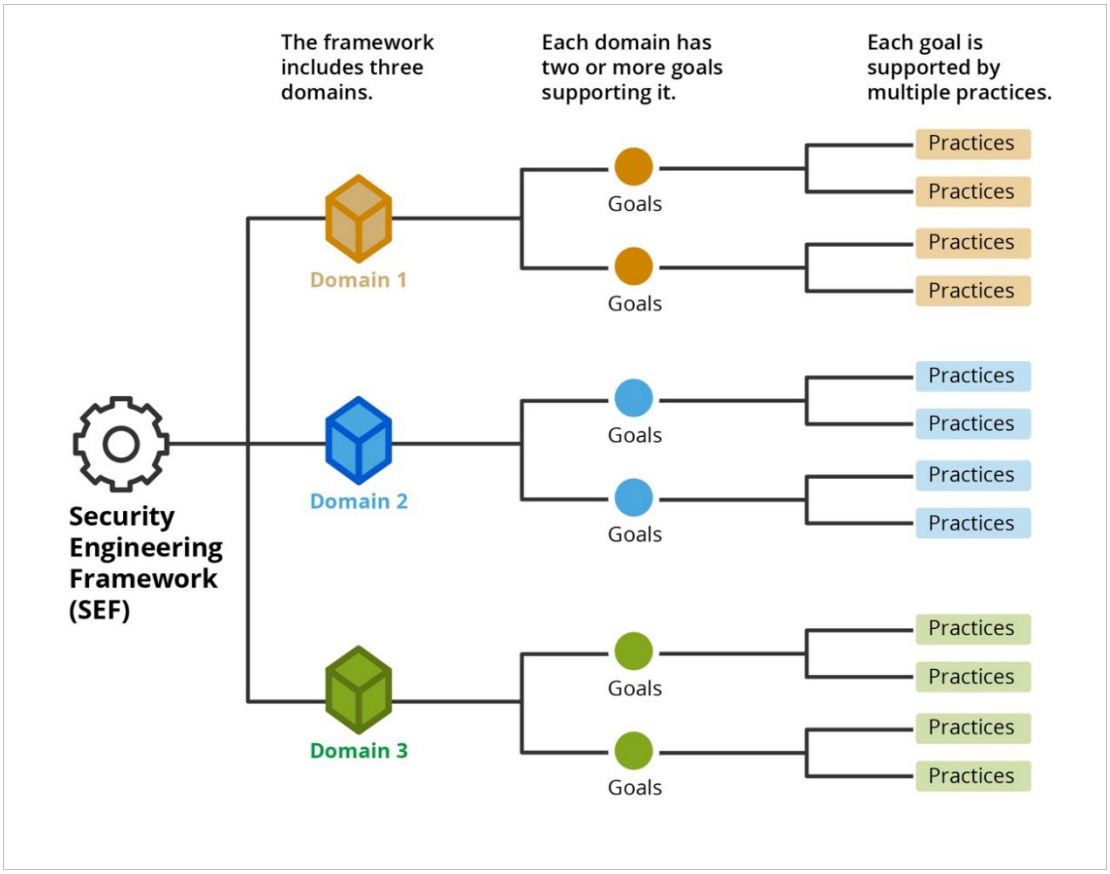

Figure 4: SEF Organization and Structure

[16] Assessments are a key aspect of security/resilience risk management. SEF practices are phrased as questions to facilitate the use of SEF content to assess a program's security/resilience practices.

The SEF comprises a total of 3 domains, 13 goals, and 119 practices. The next section describes the SEF's domains and goals.

5.1 SEF Domains and Goals

The SEF addresses security/resilience from several perspectives that are embodied in the SEF's three domains. The three core SEF domains are illustrated in Figure 5 and described thereafter.

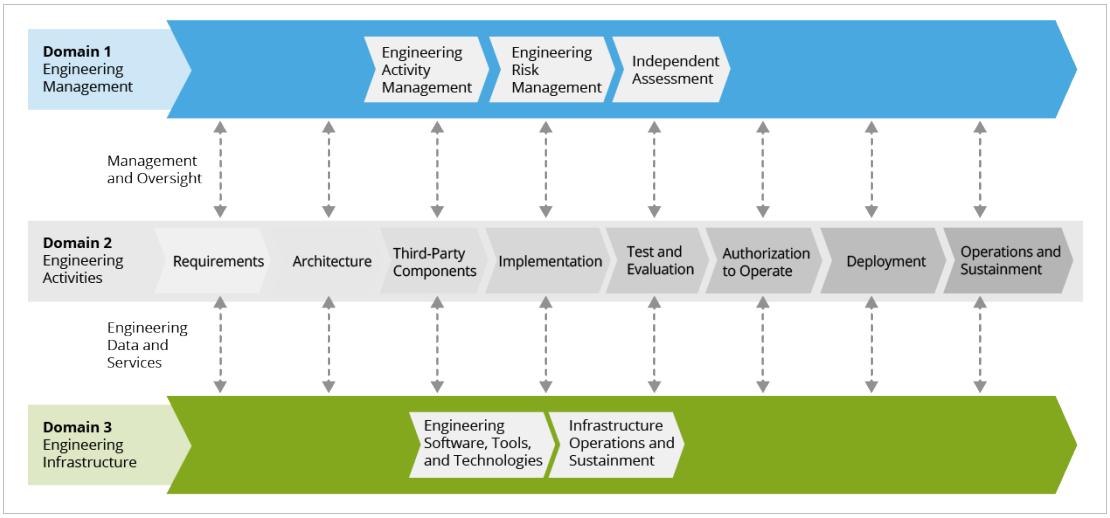

Figure 5: SEF Domains and Goals

Domain 1: Engineering Management

This domain provides a foundation for success by ensuring that security/resilience activities are planned and managed. The objective of Domain 1 is to manage security/resilience risks effectively in the system being acquired and developed.

Program and engineering managers combine their technical expertise with their business and mission knowledge to provide technical management and organizational leadership for engineering projects. Managers are tasked with planning, organizing, and directing an acquisition program's engineering and development activities. Engineering management is a specialized type of management that is needed to lead engineering or technical personnel and projects successfully. Domain 1 comprises the following three goals:

- **Goal 1.1: Engineering Activity Management.** Security/resilience engineering activities across the lifecycle are planned and managed.
- **Goal 1.2: Engineering Risk Management.** Security/resilience risks that can affect the system are assessed and managed during system design and development.
- **Goal 1.3: Independent Assessment.** An independent assessment of the program or system is conducted.

Domain 2: Engineering Activities

This domain addresses the day-to-day practices that are essential for building security/resilience into a software-reliant system. The objective of Domain 2 is to integrate security/resilience into the program's existing engineering practices. All systems lifecycles address a common set of engineering activities, beginning with requirements specification and continuing through system O&S. Domain 2 expands the focus of a program's systems lifecycle model to include security/resilience. Domain 2 comprises the following eight goals:

- **Goal 2.1: Requirements.** Security/resilience requirements for the system and its software components are specified, analyzed, and managed.
- **Goal 2.2: Architecture.** Security/resilience risks resulting from the system and software architectures are assessed and mitigated.
- **Goal 2.3: Third-Party Components.** Security/resilience risks that can affect third-party components are identified and mitigated.
- **Goal 2.4: Implementation**. Security/resilience controls are implemented, and weaknesses and vulnerabilities in software code are assessed and managed.
- **Goal 2.5: Test and Evaluation.** Security/resilience risks that can affect the integrated system are identified and remediated during test and evaluation.
- **Goal 2.6: Authorization to Operate.** The operation of the system is authorized, and the residual risk to operations is explicitly accepted.
- **Goal 2.7: Deployment.** Security/resilience is addressed in transition and deployment activities.
- **Goal 2.8: Operations and Sustainment.** Security/resilience risks and issues are identified and resolved as the system is used and supported in the operational environment.

Domain 3: Engineering Infrastructure

This domain manages security/resilience risks in the engineering, development, test, and training environments. The objectives of Domain 3 are to use software, tools, and technologies that support the program's engineering and development activities and to manage security/resilience risks in the engineering infrastructure. Engineers and developers use a variety of software, tools, and technologies to support their design and development activities. Security/resilience engineering software, tools, and technologies need to be procured, installed, and integrated with the program's existing engineering infrastructure.

The engineering infrastructure is the part of the IT infrastructure that supports engineering and development activities performed by personnel from the acquisition program, contractors, and suppliers. As a result, the engineering infrastructure can be an attack vector into the software-reliant system that is being acquired and developed. IT support teams need to ensure that they are

applying security/resilience practices when managing the engineering infrastructure to ensure that risk is being managed appropriately. Domain 3 comprises the following two goals:

- **Goal 3.1: Engineering Software, Tools, and Technologies.** Security/resilience engineering software, tools, and technologies are integrated with the engineering infrastructure.
- **Goal 3.2: Infrastructure Operations and Sustainment.** Security/resilience risks in the engineering infrastructure are identified and mitigated.

5.2 SEF Guidance

SEF domains provide the organizing structure for the framework's technical content, which is presented as a collection of goals and practices. The SEF includes in-depth guidance for all goals and practices. This guidance describes the capability represented by each goal, including its purpose, relevant context, and supporting practices. SEF guidance also provides an elaboration of each practice question that defines the key concepts and background information needed to understand the intent of the practice.

SEF guidance is presented in a standard format that is organized around the framework's 13 goals. Every SEF goal includes the following sections:

- **Goal title**—a shortened form of the goal statement (It is the heading that introduces the goal.)
- **Goal statement**—the brief statement that appears immediately after the goal title (It is the statement that clearly and completely describes the goal.)
- **Goal purpose**—a statement, positioned directly after the goal statement, that provides a concise overview of the goal's objective
- **Goal summary**—a section that encapsulates, in a few paragraphs, how the practices combine to form a picture or story that achieves the goal (It offers a high-level summary that provides enough information for the reader to understand how the practices interconnect. It may include background information, a definition, or a description of the goal's main concept.)
- **List of practice questions**—a section that lists the goal's practice questions
- **Context**—a section that provides important content while repeating details from the goal summary (Content in this section can vary significantly from goal to goal.)
- **Competencies**—a table that lists the knowledge areas and skills relevant to the goal
- **Guidance for practice questions**—a section that provides essential information (e.g., background and recommendations) to help readers understand the practice questions

SEF domains provide a structure for organizing security/resilience goals and practices. The complete SEF structure, including guidance, is provided in Part 2 of this report.

Part 2
Framework and Guidance

This part of the report makes SEF concepts accessible and actionable. The SEF comprises three domains that provide the organizing structure for the framework's goals and practices. Guidance for each SEF goal describes a capability, including its purpose, relevant context, and supporting practices. Elaborations for each practice define key concepts and background information needed to understand the intent of the practice.

Domain 1: Engineering Management

This domain provides a foundation for success by ensuring that security/resilience activities are planned and managed. The objective of Domain 1 is to manage security/resilience risks effectively in the system being acquired and developed.

Program and engineering managers combine technical expertise with business and mission knowledge to provide technical management and organizational leadership for engineering projects. Managers are tasked with planning, organizing, and directing an acquisition program's engineering and development activities. Engineering management is a specialized type of management that is needed to lead engineering or technical personnel and projects successfully. Domain 1 comprises the following three goals:

- **Goal 1.1: Engineering Activity Management.** Security/resilience engineering activities across the lifecycle are planned and managed.
- **Goal 1.2: Engineering Risk Management.** Security/resilience risks that can affect the system are assessed and managed during system design and development.
- **Goal 1.3: Independent Assessment.** An independent assessment of the program or system is conducted.

Goal 1.1: Engineering Activity Management

Security/resilience engineering activities across the lifecycle are planned and managed.

The purpose of this goal is to plan for and manage a system's security/resilience risks across the lifecycle by overseeing the execution of security/resilience engineering activities, including those performed by contractors.

Goal Summary

Engineering personnel are responsible for overseeing and managing security/resilience engineering activities across the systems lifecycle according to an appropriate model. Engineering management encompasses a wide range of tasks, including planning, managing, and monitoring engineering activities and processes; providing adequate resources for those activities and processes; training technical personnel; managing contractor activities; providing technical oversight of day-to-day technical tasks; conducting formal reviews of work products; consistently managing engineering project-level risks and issues; and escalating them as needed.

These engineering management tasks form this goal's practice questions, which are summarized in the list below and discussed in detail in the *Guidance for Practice Questions* section on page 36.

List of Practice Questions

1.1.1: Has a lifecycle model (e.g., Waterfall, Agile, DevSecOps) that includes security/resilience engineering been selected for the program?

1.1.2: Are processes for conducting security/resilience engineering activities across the lifecycle implemented, maintained, and improved?

1.1.3: Is a plan for conducting security/resilience engineering activities across the lifecycle developed and maintained?

1.1.4: Are planned security/resilience engineering activities monitored and managed?

1.1.5: Are adequate resources (e.g., funding, staffing, tools) provided to implement planned security/resilience engineering activities?

1.1.6: Is security/resilience training for technical personnel (including contractor personnel) provided as required?

1.1.7: Are security/resilience engineering activities performed by contractors managed?

1.1.8: Are security/resilience engineering activities and work products evaluated during technical reviews?

1.1.9: Are project risks and issues for security/resilience engineering activities identified and managed?

1.1.10: Is management of project risks and issues for security/resilience engineering activities performed consistently across all engineering areas and teams?

1.1.11: Are project risks and issues for security/resilience engineering activities escalated to program management and other stakeholders as appropriate?

Context

Acquisition and development programs manage engineering activities as a project. Some engineering projects focus on developing and deploying new systems, while others focus on providing new capabilities or upgrades to existing systems. The life of a typical system begins with a concept and continues through decommissioning and disposal. Over a system's lifespan, many engineering projects are initiated and performed. Each project is planned and managed to address requirements and meet stakeholder objectives.

The program's engineering lead manages engineering resources, including the personnel, budget, and schedule. The program's engineering team plays an active role in performing engineering activities across the lifecycle, including managing the activities performed by contractors.

Engineering project management includes the following four key areas, each of which plays an important role in managing security/resilience engineering activities:

- **Engineering planning and management.** This area focuses on developing and maintaining a plan for conducting engineering activities, including security/resilience engineering activities. The program manager and engineering lead select a systems lifecycle that includes security/resilience. The engineering lead then develops and documents an engineering project plan that defines the activities and processes, including security/resilience engineering activities and processes, to be performed across the lifecycle to develop the system.
- **Technical oversight of engineering activities.** The program manager and engineering lead are responsible for acquiring the resources (e.g., funding, staffing, tools) needed to carry out the engineering project plan. The engineering lead tracks progress against the plan, making adjustments as needed, and reports the project's status to the program manager. Contractors are responsible for performing many engineering and development activities, all of which require oversight (e.g., monitoring, reporting).
- **Technical reviews.** These reviews are conducted at key points throughout the systems lifecycle to establish the technical baseline, assess the system's technical maturity, evaluate technical risks, and determine the status of activities/work products. The program manager and engineering lead use these reviews to manage the program's technical effort.
- **Project risk management.** The engineering team identifies and consistently manages project risks[17] and issues that can affect security/resilience engineering activities across the lifecycle. Some of these risks might need to be escalated to the program level, where they can be more effectively managed.

[17] See Section 4.6, Project and Product Risk Management, in Part 1 for a discussion of the differences between project risk management and product risk management.

Competencies

Knowledge Areas	Skills
- lifecycle model being applied - systems engineering processes, methods, and tools - software engineering processes, methods, and tools - risk assessment processes, methods, and tools - process management and improvement methods - technology acquisition and development - security/resilience engineering principles - security/resilience risk management - security/resilience controls for systems - supply chain risk management (SCRM) practices	- selecting a systems lifecycle model that includes security/resilience engineering - developing and maintaining a plan for conducting security/resilience engineering activities - providing technical oversight for the security/resilience engineering activities that contractors perform - providing security/resilience engineering guidance to contractors and other technical personnel - monitoring progress against engineering plans - implementing and managing security/resilience processes - making technical recommendations to program/system stakeholders - reporting status information to program/system stakeholders - leading technical reviews of engineering work products - identifying and managing risks and issues that can affect security/resilience engineering activities

Guidance for Practice Questions

1.1.1: Has a lifecycle model (e.g., Waterfall, Agile, DevSecOps) that includes security/resilience engineering been selected for the program?

A systems lifecycle[18] describes the activities involved in acquiring and developing a software-reliant system from initial concept through system disposal. A lifecycle model is a conceptual representation used in project management [Pratt 2023] to describe how development activities are implemented in a given development project.

The use of lifecycle models to support system and software development can be traced back to the 1970s. Lifecycle models have evolved as technology has changed. Modern software-oriented lifecycle models (e.g., Waterfall, Agile, DevSecOps[19]) focus on the rapid, organized, and controlled delivery of high-quality software at a low cost. All of these models provide a structured means of organizing resources and managing the development of systems and software.

---

[18] Software projects may use the term *software development lifecycle* to describe the process of creating a system. Software developers use it as a guide to ensure that high-quality software is produced within the project's cost and schedule objectives. Systems engineering projects may use the term *systems development lifecycle* to describe the process of planning, creating, testing, and deploying a system. The systems development lifecycle applies to all system components, including software. The same acronym (SDLC) is used for both lifecycles, which can lead to confusion. To avoid this confusion, the Security Engineering Framework (SEF) uses the term *systems lifecycle* to describe the activities involved in acquiring and developing a software-reliant system.

[19] DevSecOps stands for Development, Security, and Operations.

One of the leading practices in modern lifecycle models involves integrating security/resilience into the selected model, helping to ensure that security/resilience is addressed across all lifecycle activities. How security/resilience is integrated into the systems lifecycle can vary based on the model selected.

1.1.2: Are processes for conducting security/resilience engineering activities across the lifecycle implemented, maintained, and improved?

Process management facilitates the predictable and efficient delivery of activities, putting an acquisition program in a position to achieve its security/resilience objectives. A key premise of process management is that organizational outcomes are highly influenced by the quality of its processes. Higher degrees of process management translate to more stable environments that produce predictable results over time and help enable mission success at lower risk.

Processes connect and integrate activities that contribute to achieving an objective or support a mission across the lifecycle. Processes can be informal and ad hoc or managed in a directed and planned manner. A systematic approach for managing processes enables an organization to anticipate and adapt to changes, allowing program personnel to manage security/resilience more effectively. By focusing on the effectiveness, efficiency, and adaptability of its processes, an acquisition program can identify and address areas that need improvement, providing potential efficiency gains and improved risk outcomes.

Effective process management requires resource investments in areas such as communication, policy and standards development, and documentation. The challenge is determining the level of investment required to achieve targeted organizational outcomes and risk levels. Process optimization is closely tied to the management of security/resilience risks.

1.1.3: Is a plan for conducting security/resilience engineering activities across the lifecycle developed and maintained?

The program's engineering lead plans and executes systems and software engineering activities. The engineering project plan identifies the tasks, personnel (for both the program and contractor), budget, and schedule for implementing the activities in the selected lifecycle model. This plan also provides a mechanism for engaging with program/system stakeholders to keep them informed of the project's progress.

Security/resilience activities are included in the engineering project plan. Otherwise, security/resilience will likely not be addressed as the system and its components are being developed. From a security/resilience perspective, the engineering project plan provides a roadmap for building security/resilience into a software-reliant system. To be effective, the plan must be flexible and updated regularly as new information is provided and circumstances change.

1.1.4: Are planned security/resilience engineering activities monitored and managed?

The engineering project plan not only identifies the tasks, personnel, budget, and schedule for implementing security/resilience engineering activities, but it also establishes objectives, a schedule, and responsibilities. Progress against the engineering project plan is monitored to ensure that the project is on track and that its objectives are being met within resource, cost, and schedule constraints.

If a project's performance deviates from the plan, corrective actions are taken. Reporting, which is highly related to monitoring, provides status information to program/system stakeholders, helping to ensure that they are engaged with the project. Security/resilience engineering activities are monitored and managed to ensure they are on track.

1.1.5: Are adequate resources (e.g., funding, staffing, tools) provided to implement planned security/resilience engineering activities?

Resources are the assets needed to produce a deliverable; they help the program progress from initial concept to system delivery. The three most critical resources for security/resilience engineering are funding, staffing, and tools. Therefore, security/resilience activities must be included in the program budget and funded adequately. This funding can be used to acquire the needed personnel and tools for performing planned security/resilience activities.

Those tasked with performing security/resilience activities must have the appropriate knowledge and skills. Because security/resilience engineering spans the entire systems lifecycle, the range of knowledge and skills required is quite extensive. Security/resilience subject matter experts (SMEs) can be expensive and difficult to find. As a result, some organizations charter a team of security/resilience SMEs who support multiple programs.

Engineers use a range of support tools when performing security/resilience activities. The type of tool varies by lifecycle activity. The cost to acquire, install, operate, and maintain a suite of security/resilience tools can be substantial.

Providing adequate resources for security/resilience activities helps ensure that security/resilience objectives are achieved. Therefore, estimates of resources, including budget and staffing allocations, must be updated periodically. Program/system stakeholders must be engaged in decisions about security/resilience resourcing to ensure that the engineering team has sufficient resources to do its job.

1.1.6: Is security/resilience training for technical personnel (including contractor personnel) provided as required?

Security/resilience engineering encompasses a wide range of activities, including requirements specification, architecture analysis, coding, code analysis, vulnerability management, penetration testing, defensive cyber operations, and risk management. Engineers and developers may also need to manage security/resilience for new and emerging technologies, including cloud computing, zero trust architectures, artificial intelligence (AI), machine learning (ML), and edge computing. The knowledge and skills required to perform security/resilience engineering activities and to manage security/resilience for new and emerging technologies are quite extensive.

An organizational training initiative can help technical personnel develop their security/resilience knowledge and skills. Training initiatives can also improve the job performance of an individual or group. Common training practices include orientations, classroom lectures, case studies, role playing, and mentoring.

1.1.7: Are security/resilience engineering activities performed by contractors managed?

Delivering a secure/resilient system is the result of executing a defined set of activities across the systems lifecycle. The engineering project plan defines the schedule of security/resilience engineering activities and assigns responsibility for completing those activities. Contractors are often responsible for performing many engineering and development activities, including requirements specification, architecture development, and implementation/coding activities.

The program's engineering team manages these contractor activities. Engineers provide guidance to contractors, monitor progress, make technical recommendations when needed, and report progress to the engineering lead and other program/system stakeholders.

1.1.8: Are security/resilience engineering activities and work products evaluated during technical reviews?

Technical reviews are events where program and engineering personnel evaluate significant achievements and assess technical maturity and risk [DAU 2023]. The program manager and engineering lead use these reviews to define and control the program's technical effort. Several technical reviews are conducted across the systems lifecycle. In defense programs, they can include a System Requirements Review (SRR), Preliminary Design Review (PDR), Critical Design Review (CDR), or Test Readiness Review (TRR).

Technical reviews of program progress are event driven and conducted when the system under development meets the criteria that are documented in the program plan. Technical reviews are important because they provide the program manager and engineering lead visibility into the progress of security/resilience activities and the quality and completeness of work products as they are developed.

1.1.9: Are project risks and issues for security/resilience engineering activities identified and managed?

A project *risk* is an uncertain event or condition that, if it occurs, produces a negative effect on a project's objectives. In contrast, a project *issue* is an obstacle or challenge that is already present (i.e., there is no uncertainty). Personnel across the acquisition program, including the engineering team, manage both project risks and issues.

From a security/resilience perspective, the engineering team identifies and manages project risks and issues that affect its ability to meet its security/resilience objectives. Conditions that can lead to project risks and issues for security/resilience activities include insufficient resources (e.g., funding, staffing, tools), scheduling difficulties, and technical problems encountered during development. The engineering team should leverage the program's methods and tools to manage these project risks and issues.

1.1.10: Is management of project risks and issues for security/resilience engineering activities performed consistently across all engineering areas and teams?

Engineering projects typically include multiple teams that work together to develop, deploy, and operate a system. In many cases, these teams report to different managers or belong to different organizations. As a result, they can implement inconsistent risk management methods and tools, which affects how risks and issues are identified, managed, and communicated across the teams on the project.

In such an environment, risk communication problems are a particular concern. Establishing a common understanding of project risks and issues is an important first step toward managing them effectively. However, diverse technical perspectives and independent management chains can be obstacles to achieving that common understanding. Engineering teams can also have competing priorities, which can lead to conflicting or inefficient risk/issue mitigations.

An integrated approach to mitigating project risks and issues requires effective governance, which is best accomplished at the program level. This governance includes establishing standards and guidelines for identifying, assessing, managing, and communicating project risks and issues across all engineering teams. Once established, all project risks and issues for security/resilience engineering activities must be managed in accordance with those standards and guidelines.

1.1.11: Are project risks and issues for security/resilience engineering activities escalated to program management and other stakeholders as appropriate?

In the context of project management, risk escalation is a formal process that transfers ownership and accountability for a project risk or issue to a higher authority or stakeholder. Escalating project risks and issues requires informing relevant program stakeholders about potential and actual problems that need their attention or intervention.

Escalation is appropriate when the engineering team and program/system stakeholders agree that a project risk or issue is outside the scope of the engineering team's responsibility or that the proposed response would exceed the engineering lead's authority. The keys to a successful escalation process are (1) establishing criteria that specify when to escalate project risks and issues and (2) defining a chain of command for escalation. Escalating project risks and issues for security/resilience engineering activities should follow the program's defined escalation process.

Goal 1.2: Engineering Risk Management

Security/resilience risks that can affect the system are assessed and managed during system design and development.

The purpose of this goal is to assess and manage security/resilience risks as the system is being designed and developed.

Goal Summary

As the system is being designed and developed, security/resilience risks must be assessed and managed. To achieve this early in the systems lifecycle, it is important to conduct periodic security/resilience risk assessments. Before conducting such an assessment, the operational environment where the system will be deployed must be defined.

Developing an understanding of the system's operational environment is critical, a key aspect of which is analyzing and documenting the mission threads/workflows that the system supports [Alberts 2014]. Another important aspect of establishing the assessment's operational scope and context is examining dataflows within the system and data exchanges with other systems.

As part of a risk assessment, risks are identified, analyzed, and prioritized; plans for managing and mitigating risks are developed, implemented, and tracked; and results of security/resilience risk management decisions are reviewed with program/system stakeholders as appropriate. System security/resilience risk assessments should start early in the systems lifecycle and be performed periodically as the system is designed and developed.

These engineering risk activities form this goal's practice questions, which are summarized in the list below and discussed in detail in the *Guidance for Practice Questions* section on page 45.

List of Practice Questions

1.2.1: Is a plan developed and documented for assessing and managing security/resilience risks for the system?

1.2.2: Are mission threads (e.g., workflows, business processes) established and maintained for the system?

1.2.3: Are dataflows within the system and data exchanges across system boundaries analyzed?

1.2.4: Are security/resilience risks for the system identified?

1.2.5: Are security/resilience risks evaluated and prioritized?

1.2.6: Are plans for mitigating security/resilience risks developed and implemented?

1.2.7: Are security/resilience risks and mitigation plans tracked?

1.2.8: Are security/resilience risk assessment and management results documented and reviewed with stakeholders?

1.2.9: Are security/resilience risk assessments performed periodically during system design and development?

Context

The engineering team manages both project and product risks across the lifecycle. Project risk management identifies and manages security/resilience risks in the *project*, including risks related to resources, cost, and schedule. Product risk management assesses and manages risks in the *product* that is being designed and developed. Within the context of the Security Engineering Framework (SEF), the product is a software-reliant system.

Engineers and developers work with security/resilience subject matter experts (SMEs) to assess and manage security/resilience risks in the system that is being acquired and developed. By addressing security/resilience risks early in the lifecycle, engineers and developers can take proactive measures to build security/resilience into the system and reduce its residual security/resilience risk when it is deployed.

The system that is undergoing a risk assessment is referred to as the *system of interest*.[20] Security/resilience risks must be managed for the system of interest across the lifecycle. This lifecycle perspective is reflected in the following SEF goals:

- This goal, Goal 1.2, Engineering Risk Management, focuses on managing security/resilience risks prior to deployment (e.g., requirements, architecture, implementation, test and evaluation [T&E]). During system deployment, the operations and sustainment (O&S) organization becomes responsible for managing security/resilience risks. Security/resilience risk information generated during system design and development (e.g., risks, mitigation plans, tracking data) is formally transitioned to the O&S organization at that time.
- Goal 2.8, Operations and Sustainment, addresses security/resilience risk management practices after system deployment.

To manage the security/resilience risks of the system and its operating environment, those risks must first be assessed. This requires developing an understanding of the operational environment. The system of interest typically supports one or more mission threads/workflows[21] and operates in a system-of-systems (SoS)[22] environment, exchanging data and services with other independently managed systems to support those mission threads/workflows. Risk assessors must therefore have a solid understanding of a system's operating environment to properly evaluate the core elements (i.e., vulnerability, threat, consequence) of the system's security/resilience risk.

---

[20] A *system of interest* is defined as the focus of the systems engineering effort. It refers to the system that is being acquired and developed and is undergoing an assessment. A system of interest includes system components, system component interconnections, and the environment in which they are placed [NIST 2020].

[21] A *workflow* is a collection of interrelated work tasks that achieves a specific result [Sharp 2008]. A workflow includes all tasks, procedures, organizations, personnel, technologies, tools, data, inputs, and outputs required to achieve the desired objectives. *Mission thread* is essentially the term that the military uses in place of *workflow*. A mission thread is a sequence of end-to-end activities and events that takes place to accomplish the execution of a military operation. The SEF uses the terms *mission thread* and *workflow* synonymously. As a result, the phrase *mission thread/workflow* is used throughout the SEF.

[22] An *SoS* is defined as a set or arrangement of interdependent systems that are related or connected (i.e., networked) to provide a given capability [Levine 2003].

During risk identification, assessors review the data that has been collected and documented for the system of interest, including data about the system and its operational environment (e.g., requirements specification, architecture documentation, workflows/mission threads supported by the system of interest) and the results of other system assessments that have been performed.

Assessors use this data to understand how the system of interest is supposed to function within its operational environment and to identify potential issues and concerns. Assessors use this knowledge to devise ways to subvert expected operational performance. In other words, they adopt the perspective of malicious actors when identifying security/resilience risks.

To a malicious actor, a successful cyber attack[23] leads to mission failure. To succeed, the malicious actor must first gain access to the system of interest (i.e., the attack target). Malicious actors often must traverse circuitous routes though an SoS environment to gain access to the system of interest. These routes are referred to as *attack paths* (also called *attack vectors*). An attack-path analysis analyzes the potential pathways that adversaries can use to access the system of interest.

After gaining access to the system of interest, the malicious actor is in a position to execute a cyber attack on that system. The cyber attack targets the information and services provided by the system of interest with the goal of causing a range of negative consequences for mission stakeholders. For a given cyber attack, the malicious actor is generally trying to produce one or more negative outcomes related to the system's confidentiality, integrity, and availability (CIA) requirements,[24] such as

- unauthorized information disclosure or information theft (confidentiality)
- modification or manipulation of information or services (integrity)
- destruction of information (availability)
- interruption of access to information or services (availability)

These outcomes are referred to as the *direct consequences* of a cyber attack because they establish the immediate results of the attack. Direct consequences often propagate to other systems within the SoS environment and to the mission threads/workflows supported by the system of interest. The impacts on other systems and mission threads/workflows are called *indirect consequences* of the attack. In a security/resilience risk assessment, assessors use indirect consequences to establish a risk's impact. They are looking to determine the extent to which a cyber attack will lead to mission degradation or mission failure.

Once assessors identify the system of interest's security/resilience risks, they must evaluate and prioritize those risks, which completes the risk assessment. After the risk assessment is complete, its results are documented and discussed with program/system stakeholders.

---

[23] A *cyber attack* is the collection of actions taken using computer networks to disrupt, disable, destroy, or maliciously control a computing environment/infrastructure; destroy the integrity of the data; or steal controlled information [NIST/DOC 2012].

[24] CIA requirements indicate what qualities of a data element are important to protect; they also provide insight into a malicious actor's cyber attack goal.

Security/resilience risk assessments should be performed periodically during system design and development. At a minimum, they should be performed at least once during each major lifecycle phase prior to deployment.

Competencies

Knowledge Areas	Skills
- technologies that are being acquired and developed - external systems and services that support the system being acquired and developed - security/resilience engineering principles - security/resilience risk management principles and methods - controls for protecting and sustaining systems and information - workflow modeling techniques - risk assessment and management principles and methods - results of security/resilience risk assessments conducted on the system, including recommended technical, physical, and administrative controls	- developing a risk management plan for the system of interest - developing models of mission threads/workflows - performing dataflow analysis - performing threat analysis - performing vulnerability assessments - performing attack-path analysis - identifying security/resilience risks - establishing acceptable risk levels for the system under evaluation - evaluating and prioritizing security/resilience risks - developing mitigation plans for security/resilience risks - preparing reports that describe risks and mitigation plans

Guidance for Practice Questions

1.2.1: Is a plan developed and documented for assessing and managing security/resilience risks for the system?

Engineering risk management begins by developing a plan for assessing and managing security/resilience risks for the system of interest. This plan defines the activities that the program intends to perform when assessing and managing security/resilience risks for the system of interest. The following are commonly found in an engineering risk management plan:

- scope of the risk management effort
- resources needed to conduct risk management activities
- roles and responsibilities required to conduct risk management
- risk management method being used, including the risk model and analytic approach
- sources of the risks being assessed
- criteria used to assess and prioritize security/resilience risks
- organizational information about security/resilience controls (e.g., control baselines, common controls)

Defense programs develop a program protection plan (PPP) to coordinate and integrate all security/resilience efforts throughout the systems lifecycle. The PPP helps ensure that the program's technology, components, and information are protected adequately [DoD 2011].

When planning a risk assessment, the initial activities are to define the assessment scope and context:

- **Risk Assessment Scope.** The scope of the assessment defines the boundary conditions of the risk assessment, establishing what it does and does not include. An assessment's scope is affected by several factors, including stakeholder needs, available resources, and time available. A scope that is too narrow can result in important objectives, assets, stakeholders, or threats being overlooked. In contrast, a scope that is too broad can waste an assessment team's time and resources and make it difficult to separate important from unimportant information (i.e., information overload). The importance of setting a proper scope for a risk assessment cannot be overstated.
- **Risk Assessment Context.** The context of the assessment defines the operational environment in which the system of interest will be deployed.

1.2.2: Are mission threads (e.g., workflows, business processes) established and maintained for the system?

A key aspect of understanding the system of interest's operational environment is analyzing and documenting the mission threads/workflows that the system supports [Alberts 2014]. A system of interest might support multiple workflows/mission threads during operations. Selecting relevant mission threads/workflows to include in the risk analysis helps to refine the scope of the assessment. When analyzing a mission thread/workflow, assessors examine the tasks, procedures, organizations, personnel, technologies, tools, data, inputs, and outputs required to achieve the desired objectives.

A mission thread/workflow can be documented in a variety of ways. Process modeling techniques use a graphical representation of a mission thread/workflow to document the individual steps performed. Other methods and tools, such as model-based systems engineering (MBSE), can be used to support the development and analysis of mission threads/workflows.

1.2.3: Are dataflows within the system and data exchanges across system boundaries analyzed?

An important aspect of defining the operational context for an assessment is examining the dataflows within the system of interest and its data exchanges with other systems (i.e., dataflow analysis). When analyzing dataflows, a variety of information is used, including system and software architectures, use cases, and external and internal interfaces. The dataflow analysis

- establishes which system components are used to store, process, and transmit data
- highlights the information and services exchanged among systems
- identifies high-priority system components and high-priority information and services

The results of the dataflow analysis are used as input to risk identification. They also provide insight into the system of interest's attack surface and potential attack paths.

1.2.4: Are security/resilience risks for the system identified?

Risk identification is the process of transforming concerns and uncertainties into distinct, tangible risks that are documented in a prescribed format. To identify security/resilience risks, assessors adopt the mindset of a malicious actor. They begin by reviewing data about the system of interest and the environment in which it operates. Relevant security/resilience data for the system of interest varies depending on the lifecycle phase in which the assessment is being performed.

Common sources of data about the system and its operational environment include the following:

- requirements specifications
- architecture documentation
- mission thread documentation
- results of previous assessments

Risk assessors look for vulnerabilities and weaknesses in the system of interest that an attacker (i.e., threat actor) can exploit. They identify potential attack paths that an attacker can use to access the system of interest. An attack-path analysis must consider both physical and cyber access to the system of interest and the broader SoS environment. Assessors also examine security/resilience risks that originate in the broader networked environment (i.e., inherited risks from external systems and services) and determine the impact of those risks on the system of interest.

For a given cyber attack, assessors evaluate direct and indirect consequences for selected mission threads/workflows. Their goal is to establish the extent to which a cyber attack will lead to mission degradation or mission failure. Relevant information is documented for each risk (e.g., vulnerabilities, weaknesses, controls, attack paths, attacks, consequences). Assessors may also formulate a risk statement (e.g., if-then statement[25]) that provides a succinct and unique description of each security/resilience risk. Using a simple notation to document risks can help streamline risk tracking.

1.2.5: Are security/resilience risks evaluated and prioritized?

Risk analysis is the process of evaluating and prioritizing security/resilience risks. During system design and development, the engineering team analyzes security/resilience risks to the system of interest. In general, three measures are associated with any risk:

- probability (likelihood of occurrence)
- impact (loss)
- risk exposure (magnitude of a risk based on probability and impact)

The engineering team's risk evaluation establishes probability, impact, and risk exposure measures for each identified security/resilience risk. These measures can be qualitative (based on low, medium, and high criteria) or quantitative (based on numerical values). In practice, most

[25] The *if* part of the statement conveys how the threat exploits a vulnerability, while the *then* part expresses the resulting consequence.

organizations implement a qualitative approach when evaluating security/resilience risks, primarily due to a lack of objective risk data.

Risk prioritization determines the order in which risks must be addressed based on their probability, impact, and risk exposure measures. Assessors must define decision-making criteria for prioritizing risks and include those criteria in the risk management plan for the system of interest.

1.2.6: Are plans for mitigating security/resilience risks developed and implemented?

Risk management methods offer multiple options for managing a risk, including the following:

- accept (take no action)
- transfer (shift to a third party)
- avoid (restructure activities to eliminate a risk)
- watch (monitor for changes)
- mitigate (take action to reduce or contain a risk)

Assessors select a management option for each risk based on its measures (i.e., probability, impact, risk exposure) and relative priority, which are established during risk analysis. High-priority security/resilience risks are generally mitigated immediately, and plans are developed to reduce or contain these risks.

Mitigation plans for security/resilience risks incorporate the following basic strategies:

- **Protect.** Reduce vulnerability to threats and minimize any consequences that might occur.
- **Detect.** Identify the occurrence of a security/resilience threat (i.e., cyber attack).
- **Respond.** Take action to counteract a detected threat (i.e., cyber attack) and minimize consequences, losses, and damages.
- **Recover.** Restore access to and functionality of a system (or systems) after a risk's consequences, losses, and damages are realized.
- **Adapt.** Enable a sustained capability to accommodate changes in a system's risk environment, including changes to threats, vulnerabilities, mission, and technologies.

Security/resilience controls are the safeguards or countermeasures prescribed for an information system or an organization to protect the CIA of the system and its information [NIST/DOC 2020]. In general, a set of security/resilience controls for reducing or containing a risk is specified in the mitigation plan. Once the plan is developed, documented, and approved, the engineering team must assign resources and implement the plan.

1.2.7: Are security/resilience risks and mitigation plans tracked?

Risk tracking monitors both risks and mitigation plans. Security/resilience risks need to be monitored because they are not static. New threats will emerge, new vulnerabilities will be identified, and technologies implemented in a system will not perform as expected. Security/resiliency requirements also change as the needs of program/system stakeholders evolve.

These changes can affect risk measures (impact, probability, risk exposure) and mitigation priorities, requiring adjustments to the system's risk profile. Significant changes affecting multiple risks may require performing another security/resilience risk assessment and updating the security/resilience risk baseline.

Risk tracking also monitors mitigation plans for efficiency and effectiveness. Engineers should monitor plans to ensure that mitigation actions are being performed as specified and that security/resilience controls are implemented correctly, operating as intended, and producing desired outcomes. A tracking system provides reports that can be shared with program/system stakeholders to update them about risk and mitigation plan status for the system of interest.

1.2.8: Are security/resilience risk assessment and management results documented and reviewed with stakeholders?

The final steps of the risk assessment and management process are to discuss information with program/system stakeholders and document the results. These discussions help stakeholders understand the security/resilience risks, priorities, and plans, and allow them to provide input. Risks and priorities might be adjusted based on these discussions. Stakeholders can also provide input to risk planning activities before plans for handling security/resilience risks are developed and implemented.

Program/system stakeholders will likely engage with the engineering team multiple times across the systems lifecycle. Building relationships with these stakeholders early in the systems lifecycle can help facilitate system design and development activities, including security/resilience risk management.

While security/resilience risk assessment results are typically documented in a formal report, risk communication involves more than simply distributing the report. It is an interactive process of exchanging risks and concerns among individuals and teams across the program [DoD 2017]. Effective risk communication strategies are customized to meet the specific interests, concerns, and needs of relevant program/system stakeholders. These strategies can include a variety of communication channels for disseminating risk information, such as the following:

- formal risk reports
- presentations
- intranet portals
- dedicated risk management platforms

Department of Defense (DoD) programs document security/resilience risk data—including threats, vulnerabilities, risks, and countermeasures—in a PPP [DoD 2011].[26]

1.2.9: Are security/resilience risk assessments performed periodically during system design and development?

A system's design evolves as it progresses across the systems lifecycle. As engineers learn about the system of interest—its requirements, design, and underlying technologies—their view of its security/resilience risks will evolve. Engineers will also learn more about the threat and risk environment in which the system will operate.

Performing system security/resilience risk assessments during design and development is a leading practice in systems and software engineering. Security/resilience risk assessments should start early in the systems lifecycle (e.g., as requirements are being developed) and be performed periodically as the system is designed and developed. These assessments should be performed at least once during each major lifecycle phase prior to deployment (e.g., requirements, architecture, implementation, T&E).

Periodic assessments enable the engineering team to assess emerging risks, evaluate controls, comply with regulations, prioritize resource allocation, support decision making, and communicate the system's security/resilience risks to stakeholders. During system deployment, the responsibility for managing security/resilience risks and information generated during system design and development is transferred to the O&S organization.

[26] For DoD programs, the compilation and dissemination of security/resilience risk information will likely include classified data. Appropriate communication channels and protocols must be implemented when risk information is classified.

Goal 1.3: Independent Assessment

An independent assessment of the program or system is conducted.

The purpose of this goal is to obtain an independent perspective of the risks and issues with the program or system.

Goal Summary

The term *independent assessment* refers to a variety of assessments that are performed by personnel who are not connected with the program or system being assessed. All independent assessments share some common elements:

- setting the scope of the assessment
- developing assessment plans and identifying limitations
- obtaining access to personnel, technology, and information
- communicating results
- tracking findings that are implemented to closure

These elements form this goal's practice questions, which are summarized in the list below and discussed in more detail in the *Guidance for Practice Questions* section on page 54.

List of Practice Questions

1.3.1: Is the scope of the independent assessment established?

1.3.2: Does a plan (including a schedule) for conducting the independent assessment exist?

1.3.3: Does the assessment team have access to the personnel who must be interviewed?

1.3.4: Does the assessment team have access to the technologies and program artifacts (e.g., schedules, contract deliverables, risk registers, reports) that must be examined?

1.3.5: Are security requirements (e.g., security clearance requirements, organizational security policies) that limit assessment activities established and communicated to the assessment team?

1.3.6: Are assessment results documented (e.g., in a formal report or presentation) and reviewed with program/system stakeholders?

1.3.7: Are assessment findings reviewed and prioritized by stakeholders?

1.3.8: Are high-priority assessment findings assigned to program and contractor personnel for implementation?

1.3.9: Is the implementation status of assessment findings tracked and reported?

Context

The purpose of an independent assessment is to (1) produce objective findings about the status of a program or system and (2) recommend ways to improve the system [Novak 2023]. Program/system stakeholders may decide to charter an independent assessment to obtain an unbiased perspective of security/resilience risks and issues. In some cases, the independent assessment might obtain specific information about the program or system to answer a specific question. In other cases, the assessment might be required by policy or regulation.

The term *independent assessment* is an umbrella term that includes a variety of assessment types. These types can be distinguished from one another based on their purpose and scope (e.g., management versus technical focus), results to be delivered (e.g., identifying problems versus solutions), and the skills needed to conduct the assessment (e.g., technical versus project management). The following are some common types of independent assessments:

- **Programmatic assessment**—a comprehensive and systematic review of an acquisition program's managerial and technical progress (This type of assessment is typically designed to identify program cost, schedule, and performance risks; formulate risk mitigation plans; and provide feedback to program stakeholders.)
- **Technical assessment**—a review that focuses on solving technical problems, answering technical questions, or supporting technical decisions (This type of assessment provides an objective technical evaluation of a system and its underlying technologies. The team members must be experts in the system's technology and domain.)
- **Compliance assessment**—a review used to establish how well a program or organization complies with a standard, policy, or regulation (From a security/resilience perspective, a compliance assessment is often used to provide an independent review of security/resilience controls selected for the system.)
- **Process assessment**—a review typically performed as part of an organizational improvement initiative (This type of assessment typically evaluates a program or organization in relation to a standard set of criteria or a model. The main purpose of a process assessment is to facilitate process improvement activities for a program or organization.)
- **Blue team assessment**—a review performed proactively to prevent problems from occurring (This type of assessment is typically scheduled well in advance of a program milestone [e.g., reviews, deliveries] to provide a program with sufficient time to implement recommended changes and prepare for the milestone successfully. Security/resilience blue teams focus on maintaining security/resilience defenses against all cyber attacks, threats, and risks [NIST/DOC 2020].)
- **Red team assessment**—a review performed when a program is experiencing significant, unanticipated problems or when program management suspects that risks or issues need immediate attention to keep the program on track (Security/resilience red teams are often chartered as adversarial exercises designed to test the susceptibility of organizational missions or business processes to compromise [NIST/DOC 2020].)

The Department of Defense (DoD) is required to perform independent technical risk assessments (ITRAs) on major defense acquisition programs (MDAPs) prior to milestone and production decisions. ITRAs provide insight into a program's technical risk that is assessed independent of the program and its chain of command. The following technical risk areas are addressed in an ITRA [DoD 2020a]:

- mission capability
- technology
- system development and integration
- modular open system approach (MOSA)
- software
- security/cybersecurity
- manufacturing
- reliability, availability, maintainability (RAM)
- sustainment

ITRAs provide DoD senior leaders with an independent view of a program's technical risks, including the maturity of its critical technologies and manufacturing processes.

A Note About *Independence*. Independent assessments are performed by personnel who are not connected with the program or system being evaluated. However, independence does not require that assessors come from an external organization. Internal assessors who are not part of the management chain for a program or system can conduct an independent assessment.

Competencies

Knowledge Areas	Skills
- independent assessment methods and tools - technologies that are being acquired and developed as part of the system being assessed - data collection methods and tools - interview techniques - technology evaluation methods and tools - data analysis methods and tools - security/resilience engineering principles - security/resilience risk management principles and methods - risk assessment and management principles and methods	- developing a plan for conducting an independent assessment - conducting interviews with program/system stakeholders - assessing program and system technologies - reviewing program artifacts (e.g., schedules, contract deliverables, risk registers, reports) - analyzing program and system data collected during the assessment - preparing a report or presentation that documents independent assessment findings - delivering a presentation of the independent assessment's findings to program/system stakeholders

Guidance for Practice Questions

1.3.1: Is the scope of the independent assessment established?

The assessment team's first step is setting the scope of the assessment. This step is essential to ensure a quality outcome of an independent assessment. The scope defines the boundary conditions of the assessment, establishing what is (and is not) included in the assessment.

An assessment's scope is affected by several factors, including stakeholder needs, available resources, and time available. As a result, the assessment scope can include an entire system, a single area or application, or an activity/function that includes multiple supporting software and hardware components.

Program/system stakeholders meet with the assessment team to discuss the goals of the assessment and define its scope. To ensure that the scope is well understood by all involved, the assessment team documents the scope and shares it with program/system stakeholders. Information about the assessment's scope can be communicated using a variety of mechanisms, including planning meetings, technical exchange meetings, and informal discussions.

1.3.2: Does a plan (including a schedule) for conducting the independent assessment exist?

The plan for the independent assessment documents the activities that the assessment team will perform to provide stakeholders with the information they requested. The assessment plan establishes an organized structure for conducting the assessment and delivering the results to stakeholders. It defines the objectives, scope, key activities, resources required, roles and responsibilities, target participants from the program, access limitations, and schedule. It also specifies the requirements for communicating results to senior stakeholders after the assessment is complete (e.g., verbal presentation, written report).

The assessment team should get input from senior program/system stakeholders when developing the plan. Engaging with stakeholders is important because it enables the assessment team to leverage stakeholders' experience and expertise. It also helps to establish stakeholder sponsorship and support for conducting the assessment.

The plan typically identifies two key roles for conducting the assessment:

- **Assessment team coordinator**—a member of the assessment team who is responsible for managing assessment logistics for the team
- **Site coordinator**—an individual from each participating location or site included in the assessment (Site coordinators are responsible for setting up data collection activities at the site and managing logistics with the assessment team coordinator.)

1.3.3: Does the assessment team have access to the personnel who must be interviewed?

Because an assessment team is not connected with the program or system being evaluated, team members typically do not have detailed knowledge of that program or system. As a result, data collection is an important part of an independent assessment. The assessment team must gain access to personnel that its members can interview to gather this information.

The assessment team normally interviews individuals who are subject matter experts (SMEs) in topics that are relevant to the assessment. Getting commitment from senior program/system stakeholders (e.g., program manager, engineering lead, contract managers) to allow program and contractor personnel to participate in interview sessions can be difficult, especially in busy or small environments. To help manage this challenge, it is useful to involve senior stakeholders who visibly support the assessment.

The assessment team coordinator works closely with each site coordinator to ensure that site personnel who will participate in interviews/workshops are identified, interviews/workshops are scheduled in a timely manner, meeting rooms and equipment are available when needed, and any unexpected events (e.g., substituting personnel in interviews/workshops) are handled appropriately.

The assessment team's flexibility can make it easier to reach program and contractor personnel. Flexibility can take several forms, including adjusting the assessment schedule, refining the assessment scope, and modifying the number or type of interview sessions. The assessment team might need to schedule interviews over several sessions and include a combination of virtual, telephone, and in-person meetings.

Finally, anonymous interviews are an important technique for eliciting program/system issues because individuals may hesitate to speak openly about risks and issues with program/system personnel. Anonymity provides individuals an opportunity to openly discuss risks and issues and provide important information and insights to the assessment team [Novak 2023].

1.3.4: Does the assessment team have access to the technologies and program artifacts (e.g., schedules, contract deliverables, risk registers, reports) that must be examined?

The assessment team must be given access to the technologies and program artifacts (e.g., schedules, contract deliverables, risk data, reports) that will be evaluated. The assessment team coordinator should work with each site coordinator to ensure that the assessment team has the necessary access.

Many issues can hinder an assessment team's ability to access technologies and program artifacts. For example, some program/system stakeholders might believe that the assessment will adversely impact the program's development activities and schedule. In some cases, third parties that manage program/system technologies might not be contractually required to participate in an

independent assessment. In these cases, the assessment team might be unable to examine some third-party technologies and associated artifacts.

The team must work with senior program/system stakeholders (e.g., program manager, engineering lead, contractor managers) to resolve access issues related to technologies and artifacts. Contractual issues involving third parties can be difficult to resolve. As a result, the scope of the assessment might need to be adjusted.

1.3.5: Are security requirements (e.g., security clearance requirements, organizational security policies) that limit assessment activities established and communicated to the assessment team?

The assessment team must adhere to the security policies established by the acquisition program and its contractors. Security requirements that limit assessment activities and restrict access to personnel, technologies, and artifacts should be documented and communicated broadly. To conduct an effective independent assessment, it is essential to identify and manage assessment constraints. Key areas that must be addressed are non-disclosure agreements (NDAs), security clearances, and permission for technology evaluations.

NDAs prevent those signing the agreement from disclosing sensitive and confidential information. Independent assessors might be asked to sign NDAs and agree not to disclose sensitive information. Assessors must also have appropriate security clearances to access program and system data, technologies, and artifacts.

Any issues regarding NDAs and security clearances must be resolved early in the assessment process (e.g., during assessment planning activities). If the assessment team plans to evaluate technologies using vulnerability scanning or penetration testing tools, it must ensure that it has all the required permissions and follows the prescribed security policies of the program and its contractors. All affected program/system stakeholders must be alerted when an assessment team will evaluate technologies using vulnerability scanning or penetration testing tools.

1.3.6: Are assessment results documented (e.g., in a formal report or presentation) and reviewed with program/system stakeholders?

An independent assessment team must effectively communicate its findings to the assessment's sponsor and other stakeholders that the sponsor specifies. The assessment findings should include strengths and weaknesses, root causes of problems, current and potential impacts, and actionable recommendations. All findings should be substantiated, and all recommendations should be clearly written.

Two mechanisms are commonly used to communicate an independent assessment's findings: formal reports and presentations. Submitting a formal report or delivering a presentation brings the independent assessment to a close.

When conducting an independent assessment, a key guideline is to avoid surprising program/system stakeholders with unexpected information. The assessment team should provide stakeholders

with ongoing updates about assessment activities and results. It should engage with stakeholders to gather their input and knowledge, and, when results are available, get their feedback.

A formal report or presentation is the primary channel for communicating the final assessment results. However, additional channels (e.g., intranet portals, web forums) can be used to communicate assessment results to a broader audience. Effective communication requires the assessment team to customize information to meet the specific interests, concerns, and needs of the target audience.

1.3.7: Are assessment findings reviewed and prioritized by stakeholders?

An independent assessment provides program/system stakeholders with a more accurate understanding of the system's status, risks, and issues and puts them in a position to implement meaningful improvements.

Improvement ultimately requires program and contractor personnel to address the assessment findings. However, before implementation, program/system stakeholders must review and prioritize the assessment findings. Stakeholders who have a broad perspective of the program/system (e.g., program manager, engineering lead, contractor managers) will lead this review/prioritization. These stakeholders should engage with SMEs as needed to gain an in-depth understanding of selected issues.

1.3.8: Are high-priority assessment findings assigned to program and contractor personnel for implementation?

After senior stakeholders review and prioritize the findings of the independent assessment, the program manager must assign responsibility for addressing the high-priority findings. Depending on the structure of the program and the complexity of a given issue, multiple individuals or teams might be assigned to address a given finding. As with most management activities, it is a good practice to document and communicate roles and responsibilities, actions to be taken, the schedule, and resources needed to address all high-priority findings.

1.3.9: Is the implementation status of assessment findings tracked and reported?

Individuals or teams assigned to address findings must develop and implement task plans. These individuals or teams are also responsible for reporting the status of addressing findings to program/system stakeholders. A status report should (1) document progress toward technical, schedule, and cost objectives and (2) record risks and issues that could derail progress.

Status tracking is an important management activity, so the program should already have a system in place for tracking and reporting program information, such as status updates and action items. The implementation status of assessment findings can be added to existing tracking and reporting systems. Program and contractor personnel are familiar with existing systems and the reports that are provided.

Domain 2: Engineering Activities

This domain addresses the day-to-day practices that are essential for building security/resilience into a software-reliant system. The objective of Domain 2 is to integrate security/resilience into the program's existing engineering practices. All systems lifecycles address a common set of engineering activities, beginning with requirements specification and continuing through system operations and sustainment (O&S). Domain 2 expands the focus of a program's systems lifecycle model to include security/resilience. Domain 2 comprises the following eight goals:

- **Goal 2.1: Requirements**. Security/resilience requirements for the system and its software components are specified, analyzed, and managed.
- **Goal 2.2: Architecture**. Security/resilience risks resulting from the system and software architectures are assessed and mitigated.
- **Goal 2.3: Third-Party Components**. Security/resilience risks that can affect third-party components are identified and mitigated.
- **Goal 2.4: Implementation**. Security/resilience controls are implemented, and weaknesses and vulnerabilities in software code are assessed and managed.
- **Goal 2.5: Test and Evaluation**. Security/resilience risks that can affect the integrated system are identified and remediated during test and evaluation.
- **Goal 2.6: Authorization to Operate**. The operation of the system is authorized, and the residual risk to operations is explicitly accepted.
- **Goal 2.7: Deployment**. Security/resilience is addressed in transition and deployment activities.
- **Goal 2.8: Operations and Sustainment**. Security/resilience risks and issues are identified and resolved as the system is used and supported in the operational environment.

Goal 2.1: Requirements

Security/resilience requirements for the system and its software components are specified, analyzed, and managed.

The purpose of this goal is to specify the security/resilience capabilities that the system and its software components should provide.

Goal Summary

Security/resilience requirements specify the capabilities or needs that a system and its software components must satisfy to mitigate security/resilience risks. A program should have a defined method for eliciting, categorizing, and prioritizing security/resilience requirements at both the system and software levels. Program personnel should inspect the security/resilience requirements to ensure they are consistent, nonredundant, and complete. They should also make sure that security/resilience requirements at the system and software levels do not conflict.

Since it is important to be able to trace requirements, it is critical that organizations structure them (e.g., using a requirements traceability matrix [RTM]). Establishing quality criteria for requirements helps organizations assess and monitor processes, practices, methods, and work products to ensure they are consistent with defined standards or models. Periodically reviewing and managing these requirements helps ensure that, when implemented, they will meet the needs of system users and address the system's operational mission.

These requirements activities form this goal's practice questions, which are summarized in the list below and discussed in detail in the *Guidance for Practice Questions* section on page 61.

List of Practice Questions

2.1.1: Are security/resilience requirements for the system and its software components elicited, categorized, and prioritized?

2.1.2: Are inspections of security/resilience requirements performed to ensure their completeness and sufficiency?

2.1.3: Are security/resilience requirements structured to ensure that their traceability is maintained?

2.1.4: Are quality criteria for security/resilience requirements established?

2.1.5: Are reviews conducted periodically to determine whether security/resilience requirements meet established quality criteria?

Context

Security protects information by preventing, detecting, and responding to attacks. *Resilience* focuses on the ability to anticipate, withstand, recover from, and adapt to adverse conditions, stresses, attacks, or compromises on systems that use or are enabled by cyber resources. *Requirements* document the necessary attributes, capabilities, characteristics, and qualities of the system that benefit stakeholders.

Requirements definition is a critical part of systems development. It begins early in the lifecycle, and requirements must be managed continuously throughout a system's acquisition and

development. Since a system's security/resilience requirements are significantly influenced by threat and risk information, early integration of those requirements into the systems lifecycle is a cost-effective way to mitigate security/resilience risks [NIST/DOC 2011].

Security/resilience requirements specify the essential capabilities that the system and its software components must satisfy to mitigate security/resilience risks. Security/resilience requirements are essentially statements that describe the system and software functionality necessary to protect information and recover from attacks. They provide the foundation for proactively managing security/resilience risks across the systems lifecycle—from the earliest stages of conceptual design through system operations and sustainment (O&S). Therefore, security/resilience requirements are critically important and must be properly documented and constructed to ensure consistency, nonredundancy, and completeness.

Different types of requirements are developed across the systems lifecycle, ranging from high-level and concept-focused requirements to highly technical ones. Common types of requirements include business or mission requirements, user requirements, system requirements, and software requirements. Security/resilience must be considered in each type of requirement, especially in system and software requirements.

Requirements analysis determines which security/resilience capabilities the system and software should provide. Decisions about which security/resilience capabilities to implement should be driven by an assessment of risk tolerance and system/component criticality.

Missing, incomplete, ambiguous, or conflicting requirements can lead to security/resilience weaknesses in system and software architectures. Therefore, eliciting, categorizing, and prioritizing security/resilience requirements can help minimize security/resilience weaknesses. Periodically reviewing and managing these requirements helps ensure that they meet the needs of system users and address the system's mission objectives over time.

Developing a secure/resilient system involves addressing requirements from two key perspectives: risk mitigation and compliance. For risk mitigation, engineers analyze the security/resilience risks to the system, its software components, and its mission, and they establish requirements that appropriately mitigate those risks. For compliance, engineers derive security/resilience requirements from applicable policies, standards, laws, and regulations.

Collaborative effort among the program manager, engineers, and security/resilience subject matter experts (SMEs) across multiple mission and support areas is required to specify and manage security/resilience requirements and ensure adequate controls are established. That said, a list of security/resilience controls is not the same as a requirements specification. *Security/resilience requirements* specify the capabilities and needs that a system and its software components must satisfy to mitigate security/resilience risks; *security/resilience controls* are a means of satisfying security/resilience requirements.

Competencies

Knowledge Areas	Skills
- standards, laws, regulations, and policies governing security/resilience in acquisition and development - technologies that are being acquired and developed - security/resilience threats and vulnerabilities - controls for protecting systems, software, and information - risk assessment - abuse and misuse cases	- documenting business processes/mission threads - documenting security/resilience misuse and abuse cases - translating governance requirements and risk assessment results into security/resilience requirements - inspecting security/resilience requirements - documenting security/resilience requirements in an appropriate format

Guidance for Practice Questions

2.1.1: Are security/resilience requirements for the system and its software components elicited, categorized, and prioritized?

Security/resilience requirements specify the capabilities that the system and its software components must satisfy to mitigate security/resilience risks and comply with applicable policies, standards, laws, and regulations. These requirements can be high level (e.g., defining confidentiality, integrity, and availability [CIA] parameters for a system and its software components) or detailed and specific (e.g., specifying the types of encryption protocols and authentication mechanisms that will be included in the system and software).

A program should define a method for eliciting, categorizing, and prioritizing security/resilience requirements for a system and its software components. The method should include the following:

- techniques for eliciting requirements (e.g., structured interviews, misuse/abuse cases, risk analysis, facilitated meetings)
- techniques for categorizing requirements by level (e.g., system, software) and importance (e.g., essential, nonessential)
- techniques for prioritizing which requirements to implement and in what order

2.1.2: Are inspections of security/resilience requirements performed to ensure their completeness and sufficiency?

Requirements inspection is one of the most important elements of creating accurate and verifiable security/resilience requirements. Inspections can be done at varying levels of formality, from error/defect inspections to code reviews. The primary goal of an inspection method is to identify defects in the requirements, such as missing requirements, ambiguous requirements, inconsistent requirements, or flawed assumptions.

2.1.3: Are security/resilience requirements structured to ensure that their traceability is maintained?

The impacts of system and software changes must be traced to related requirements and system and software architectures. Therefore, requirements traceability is an important part of managing changes to a system.

For requirements, three types of bidirectional traceability are typically maintained:

- relationships between requirements and their sources (e.g., stakeholders who proposed them)
- dependencies among requirements
- relationships between requirements and the related system and software architectures

An RTM can help an organization logically capture requirements, establish where they came from, determine what standards apply, and document changes to them as development activities progress through the systems lifecycle. The RTM should contain attributes that maintain the three types of traceability information for each requirement. The RTM also helps identify overlaps or conflicts among different sets of requirements. Traceability is important for all types of requirements, including security/resilience requirements.

2.1.4: Are quality criteria for security/resilience requirements established?

Quality criteria are used to assess and monitor processes, practices, methods, and work products to ensure they are consistent with defined standards or models. Program personnel review work products and activities to verify that they meet predefined quality criteria. A properly constructed security/resilience requirement must be unambiguous, testable (i.e., verifiable), clear, correct, and understandable. The set of security/resilience requirements for both the system and its software components should be consistent, nonredundant, and complete (i.e., there are no missing or partially formed requirements).

2.1.5: Are reviews conducted periodically to determine whether security/resilience requirements meet established quality criteria?

A review is a structured process where senior stakeholders from the program, contractors, and system users walk through security/resilience requirements for both the system and software in detail. They analyze and verify that requirements meet predefined quality criteria and are current. These requirements should be reviewed periodically to do the following:

- identify any errors or omissions
- determine whether updates or changes are required
- confirm that the overall quality of the requirements is acceptable
- communicate with program/system stakeholders to confirm system and software requirements

Goal 2.2: Architecture

Security/resilience risks resulting from the system and software architectures are assessed and mitigated.

The purpose of this goal is to assess and mitigate security/resilience risks resulting from weaknesses in the system and software architectures.

Goal Summary

A *system architecture* documents the structure of system components, their relationships, and the principles and guidelines governing their design and evolution over time. It is common for a system architecture to include a functional architecture and a physical architecture. A *software architecture* is the set of structures needed to reason about the system, which comprises software elements, relations among them, and properties of both [Clements 2010]. It documents the high-level design of a software component. The software architecture represents the design decisions related to the overall structure and behavior of a system's software components.

During the architecture phase of the systems lifecycle, architects and engineers select an initial set of security/resilience controls to incorporate into the system and its software components. This initial set of controls is then tailored and adjusted based on an assessment of security/resilience risks, which must be mitigated and tracked. Follow-on analyses are conducted, including an architecture tradeoff analysis and attack-path analysis. These analyses enable architects and engineers to (1) identify and prioritize design decisions across multiple system attributes and (2) minimize the attack surface.

To resolve issues with security/resilience features, a cross-check of the system and software architectures is performed to confirm how these features are allocated across system components. As the system and software architectures change, security/resilience requirements are updated. Stakeholders are informed about risks that were identified in the architecture tradeoff analysis and whether risks in the system and software architectures are mitigated satisfactorily.

These architecture activities form this goal's practice questions, which are summarized in the list below and discussed in detail in the *Guidance for Practice Questions* section on page 65.

List of Practice Questions

2.2.1: Are security/resilience controls defined and documented for the system and its software components?

2.2.2: Is a security/resilience risk assessment of the system and software architectures performed?

2.2.3: Are security/resilience risks in the system and software architectures mitigated and tracked?

2.2.4: Is an architecture tradeoff analysis of quality attributes, including security/resilience, performed?

2.2.5: Are security/resilience risks resulting from architecture tradeoffs communicated to stakeholders?

2.2.6: Is the attack surface minimized based on the results of an attack-path analysis?

2.2.7: Is a cross-check of the system and software architectures performed to resolve issues or inconsistencies in security/resilience features?

2.2.8: Are security/resilience requirements updated periodically to reflect security/resilience changes to the system and software architectures?

2.2.9: Are reviews conducted with stakeholders to ensure that security/resilience risks resulting from the system and software architectures are mitigated sufficiently?

Context

Design weaknesses are defects or flaws in the system and software architectures that can be exploited when the architectures are implemented. It is important to address design weaknesses as soon as possible because it is typically not easy to correct them after a system is deployed. Since these weaknesses are difficult to correct, systems are often allowed to operate under a high degree of residual security/resilience risk, putting their operational missions in jeopardy.

Reducing the number of weaknesses in the system and software architectures is an important part of acquiring and developing systems with an acceptable level of security/resilience risk. Risk management is a foundational aspect of a program's security/resilience engineering activities. It is not a one-time activity; risk management should be performed periodically across the systems lifecycle. A program's risk assessment and management activities help ensure that high-priority security/resilience risks are mitigated and relevant compliance requirements are met.

Architecture risk analysis/threat modeling should be performed on both the system and software architectures. Many programs analyze only the system architecture for security/resilience risks. This is problematic because some design weaknesses originate in the lower level design aspects of the software architecture. Weaknesses in the software architecture may not be apparent/visible when performing a system-level architecture analysis. As a result, attackers may be able to exploit design weaknesses in the software architecture if those weaknesses are not corrected.

Competencies

Knowledge Areas	Skills
- technologies that are being acquired and developed - external systems and services that support the system being acquired and developed - methods for performing architecture risk analysis - security/resilience risk management principles and methods - controls for protecting and sustaining systems and information - results of risk management (e.g., NIST RMF[27]) assessments conducted on the system and its software components - results of security/resilience risk assessments, including recommended technical, physical, and administrative controls	- applying methods, standards, and approaches to describe, analyze, and document the system and software architectures - analyzing candidate architectures, allocating security services, and selecting security mechanisms - evaluating the system and software architectures to determine whether security/resilience controls and mitigations are adequate - integrating appropriate security technologies into the proposed system and software architectures - advising program/system stakeholders (e.g., authorizing official [AO], engineering lead, program manager) about security/resilience issues affecting the system and software architectures

[27] The NIST RMF is defined in the National Institute of Standards and Technology Special Publication 800-37, *Risk Management Framework for Information Systems and Organizations*.

Guidance for Practice Questions

2.2.1: Are security/resilience controls defined and documented for the system and its software components?

During the architecture phase of the systems lifecycle, an initial set of security/resilience controls to be incorporated into the system are identified. This initial set of controls is then tailored and adjusted based on an assessment of security/resilience risks.

Security/resilience controls include technical, physical, and administrative controls. Architects and engineers start the control-selection process by analyzing adverse impacts resulting from the loss of confidentiality, integrity, and availability (CIA) of information. The resulting impact values and the system's security/resilience requirements become the input to the control-selection process.

Architects and engineers generally use one of two basic approaches for selecting controls [NIST/DOC 2018]:

- **Baseline control selection** uses predefined sets of controls, called control baselines, as the starting point for selecting security/resilience controls.
- **Organization-generated control selection** does not leverage a predefined set of controls. Instead, an organization uses its own processes to select security/resilience controls. Organizations tend to use their own processes when the system being designed and developed is highly specialized (e.g., weapons system, medical device), has a limited purpose or scope, or must operate in a unique threat environment.

Using either of these approaches, the initial set of controls includes both system controls and common controls:

- **System controls** are security/resilience safeguards or countermeasures that are implemented at the system level.
- **Common controls** are security/resilience controls that are inherited by the system.

After the initial set of controls is selected, it must be documented and reviewed with program/system stakeholders to get their input. During architecture development, engineers select a subset of controls (primarily technical controls) that will be incorporated into the system and its software components. Firewalls, intrusion detection systems (IDSs), encryption, and identification and authentication mechanisms are examples of technical controls. The selected controls are allocated to the system and software architectures as appropriate.

2.2.2: Is a security/resilience risk assessment of the system and software architectures performed?

A security/resilience risk assessment of the system and software architectures identifies weaknesses (i.e., defects or flaws) and evaluates the risks they pose to the mission. Weak authentication protocols (e.g., single-factor authentication) and unencrypted communication channels are

examples of design weaknesses. The assessment also examines security/resilience risks resulting from the broader networked environment (i.e., inherited risks from external systems and services).

During a risk assessment, the following activities are performed:

- **Risk identification** is the process of identifying the system's high-priority components, establishing interactions among those components, identifying weaknesses in the system and software architectures that malicious threat actors could exploit, and documenting risks associated with the identified weaknesses.
- **Risk analysis** is the process of evaluating and prioritizing security/resilience risks. The information included in a risk analysis becomes more detailed as the program moves through the lifecycle, enabling a more comprehensive analysis of the system over time. As more details about a system become known, new risks can be identified, evaluated, and addressed. Risk analysis comprises the following sub-activities:
  - **Risk evaluation** establishes probability, impact, and risk exposure measures for each identified security/resilience risk.
  - **Risk prioritization** establishes the order in which risks should be addressed based on their probability, impact, and risk exposure measures.

A program should define a method or approach for performing security/resilience risk assessments of the system and software architectures.[28] Architects and engineers can use the results of these assessments to address design weaknesses and implement recommended security/resilience controls.

2.2.3: Are security/resilience risks in the system and software architectures mitigated and tracked?

Whether or not a system meets its mission objectives is a crucial measure of a system's success, and security/resilience risks can disrupt the potential for mission success. Risk management options for addressing risks include the following:

- accept (take no action)
- transfer (shift to a third party)
- avoid (restructure activities to eliminate a risk)
- watch (monitor a risk for changes)
- mitigate (take action to reduce or contain a risk)

Assessors select a risk management option based on the risk's measures (i.e., probability, impact, risk exposure) and their relative priority, both of which are established during the risk assessment.

[28] Goal 1.2, Engineering Risk Management, defines a set of practices for performing a security/resilience risk assessment. These practices can be applied at different points in the systems lifecycle. The information analyzed during the risk assessment becomes more detailed as the system moves through the lifecycle, enabling a more comprehensive analysis of the system and software architectures over time.

High-priority security/resilience risks are generally mitigated. Plans for mitigating security/resilience risks incorporate the following basic strategies:

- **Protect.** Reduce vulnerability to threats and minimize any consequences that might occur.
- **Detect.** Identify the occurrence of a security/resilience threat (i.e., cyber attack).
- **Respond.** Take action to counteract a detected threat (i.e., cyber attack) and minimize consequences, losses, and damages.
- **Recover.** Restore access to and functionality of a system (or systems) after a risk's consequences, losses, and damages are realized.
- **Adapt.** Enable a sustained capability to accommodate changes in a system's risk environment, including changes to threats, vulnerabilities, mission, and technologies.

In general, a set of security/resilience controls for reducing or containing a risk will be specified in a mitigation plan. Mitigation actions include selecting or tailoring security/resilience controls, modifying the system or software architecture, and making changes to the system's operating procedures.

The status of security/resilience risks and mitigation plans must be tracked over time. A tracking system provides reports of the remediation status of identified weaknesses that can be shared with program/system stakeholders.

2.2.4: Is an architecture tradeoff analysis of quality attributes, including security/resilience, performed?

Quality attributes are functional and nonfunctional requirements that are used to evaluate system performance [Hilburn 2023]. Examples of quality attributes are performance, security/resilience, reliability, interoperability, usability, portability, maintainability, and scalability. Depending on the system being designed and developed, some quality attributes are more important than others.

Unfortunately, it is impossible to optimize a system for all quality attributes that are important to program/system stakeholders. As a result, architects and engineers must conduct a tradeoff analysis of the quality attributes [Hilburn 2023]. Using architecture tradeoff analysis helps identify and prioritize design decisions across multiple quality attributes, including security/resilience.

With this type of analysis, tradeoffs are evaluated among candidate design options, documented clearly, and tracked as part of a program's risk management activities. Each candidate design option's impact on the system's quality attributes is evaluated, and the option that best addresses the system's requirements overall is selected.

Tradeoff analysis considers the potential impact of security/resilience in relation to the potential impact of other quality attributes. By conducting an architecture tradeoff analysis, architects and engineers can make informed decisions about the system and software architectures, ensuring that risks are kept within an acceptable tolerance across the system's quality attributes.

2.2.5: Are security/resilience risks resulting from architecture tradeoffs communicated to stakeholders?

Architects and engineers can make informed decisions and manage security/resilience risks across quality attributes based on architecture tradeoff analysis. However, these risks cannot be addressed if they are not communicated to and understood by program/system stakeholders. Related decisions and their rationale must therefore be shared with stakeholders through channels such as technical interchange meetings that include system architects, system engineers, software engineers, and program/system stakeholders. These meetings can be initiated early in architecture development and held periodically to discuss design options, recommendations, and decisions.

2.2.6: Is the attack surface minimized based on the results of an attack-path analysis?

An attack surface is the set of points on the boundary of a system, system component, or environment where an attacker can try to enter, cause an effect on, or extract data from that system, system component, or environment [NIST 2020]. An attack path (i.e., attack vector) is a pathway or method that an attacker uses to access a system and exploit weaknesses and vulnerabilities in the system and its software components. An attack-path analysis evaluates the potential pathways that adversaries can use to access a system and its software components.

The result of an attack-path analysis can be used to reduce the attack surface by limiting or removing unnecessary access points or services that are identified as weaknesses or vulnerabilities. This reduction in the attack surface can include removing or disabling certain ports, protocols, or services that are not required for the system's operation. The reduction of the attack surface can help ensure that a system and its software components remain secure and resilient against potential threats.

2.2.7: Is a cross-check of the system and software architectures performed to resolve issues or inconsistencies in security/resilience features?

A system's design should confirm how architectural characteristics and design properties (e.g., security/resilience properties) are allocated and partitioned across the system and its components. The system and software architectures are not static constructs; over time, they evolve as the system and its software components mature across the systems lifecycle. The system and software architectures should be cross-checked to resolve issues or inconsistencies in the implementation of the system's security/resilience features.

2.2.8: Are security/resilience requirements updated periodically to reflect security/resilience changes to the system and software architectures?

Security/resilience risks are not static. Changes that affect the system sometimes require modifications to its system or software architectures. The impacts of architectural changes can be traced back to their corresponding requirements. In some instances, architectural changes help architects and engineers identify gaps in the system or software security/resilience requirements. When

appropriate, security/resilience requirements should be updated to reflect changes to the system and software architectures. Requirements traceability is an important part of managing these changes.

2.2.9: Are reviews conducted with stakeholders to ensure that security/resilience risks resulting from the system and software architectures are mitigated sufficiently?

Stakeholders, senior representatives from the program, contractors, and users carefully walk through the system and software architectures and conduct the following reviews:

- **The Preliminary Design Review (PDR)** ensures the planned technical approach will meet the overall system requirements. It establishes whether the basic system architecture is complete and provides confidence to proceed with the detailed design [DoD 2022e].
- **The Critical Design Review (CDR)** ensures the implemented design meets the system and software requirements. It provides stakeholders with evidence that the system and its software components have a reasonable expectation of satisfying performance requirements within cost and schedule constraints [DoD 2022e]. CDRs for software components establish whether the software architecture is complete and ready to be implemented (i.e., coded and unit tested).
- **Technical interchange meetings** between developers and other relevant program/system stakeholders can be held periodically across the lifecycle to discuss security/resilience risks and issues related to the system and software architectures.[29]

From a security/resilience perspective, these activities provide opportunities to do the following:

- evaluate the system and software architectures to identify design weaknesses
- confirm that the overall quality of the system and software architectures is acceptable
- communicate and confirm architectural decisions and tradeoffs with program/system stakeholders

[29] *Technical interchange meeting* is a general term that refers to a meeting between relevant program/system stakeholders to discuss technical information about a system's design and development. Technical interchange meetings are scheduled periodically across the systems lifecycle.

Goal 2.3: Third-Party Components

Security/resilience risks that can affect third-party components are identified and mitigated.

The purpose of this goal is to develop a bill of materials (BOM), including a software bill of materials (SBOM), for the system and ensure that operational security/resilience risks in third-party components (TPCs) are managed over time.

Goal Summary

TPCs are custom and off-the-shelf components (i.e., hardware, software, firmware, services) that contractors, suppliers, and other external organizations provide. When evaluating and selecting TPC providers, the program must ensure that the selected providers use appropriate security/resilience development practices.

To manage TPCs effectively, engineers and developers must use a scheme to identify each TPC included in the system and establish a repository for tracking their use. These components are listed in BOMs and SBOMs to help programs identify risks.

To track software licenses, defects, vulnerabilities, and dependencies, the provenance of software components must be established. The program must also establish a process for monitoring TPC vulnerabilities and communicating about them, including a way to assess each TPC for the operational risks it might introduce and prioritize the related mitigations.

These TPC-related activities form this goal's practice questions, which are summarized in the list below and discussed in detail in the *Guidance for Practice Questions* section on page 71.

List of Practice Questions

2.3.1: Is a scheme that uniquely identifies each third-party component implemented?

2.3.2: Is a repository that tracks the use of third-party components in systems implemented and maintained?

2.3.3: Are third-party components that are used in the system identified and documented to create a bill of materials/software bill of materials?

2.3.4: Is the provenance of software components (including origin, development history, and change history) established and tracked?

2.3.5: Are third-party component providers (e.g., contractors, suppliers) evaluated and selected based on their use of secure/resilient development practices?

2.3.6: Is each third-party component's operational risk assessed?

2.3.7: Is each third-party component monitored for vulnerabilities and available updates?

2.3.8: Are third-party components prioritized for mitigation based on operational risk?

Context

Using TPCs for system development and operation has become a standard practice in acquisition programs and the greater acquisition community. This use provides acquisition programs with cost-effective access to a broad network of contractors and suppliers that have specialized skills, components, and infrastructures. At the same time, these third-party relationships create dependency risks that the program must manage in the context of its overall risk management strategy.

TPCs include custom and off-the-shelf components that contractors, suppliers, and other external organizations provide. Since engineers and developers often integrate these components into the system and its components, TPCs should be monitored for issues, vulnerabilities, and updates. This monitoring helps organizations quickly identify and address risks from defective or vulnerable TPCs. Managing security/resilience risks from TPCs is an important aspect of supply chain risk management (SCRM).

Since software TPCs can include government-off-the-shelf (GOTS) software, commercial off-the-shelf (COTS) software, open source software (OSS), and custom-developed software, it is important for organizations to use an SBOM. It lists the system's software components and their versions and helps organizations manage software components effectively.

Using TPCs is an important aspect of systems security engineering and software assurance. However, programs must manage TPC risks and proactively identify new/unknown vulnerabilities before they can be exploited.

Competencies

Knowledge Areas	Skills
- SCRM practices - format and content of SBOMs - vulnerability management - relevant vulnerability data sets - security/resilience risk management	- applying white-box testing (e.g., static code analysis) methods, tools, and techniques to identify vulnerabilities - applying black-box testing (e.g., dynamic code analysis, vulnerability scanning) methods, tools, and techniques to identify vulnerabilities - conducting vulnerability scans and recognizing vulnerabilities in system components - remediating or correcting identified coding weaknesses and vulnerabilities

Guidance for Practice Questions

2.3.1: Is a scheme that uniquely identifies each third-party component implemented?

TPCs include hardware, software, firmware, and services that are provided by third parties (e.g., contractors, suppliers, external organizations) and integrated into the system. To manage TPCs effectively, engineers and developers must uniquely identify each TPC included in the system. While the specifics of various identification schemes may vary, the main objective of using such a scheme is to facilitate the ongoing management of all system components and the risks they can pose to the system's mission.

For software components, an acquisition program must implement a common scheme (i.e., structure) that explicitly identifies the software components included in the system. The data scheme should capture the baseline information about each software component that will be tracked and maintained. Examples of baseline information for software components include supplier name, component name, version, license information, dependency information, author of the SBOM data, and a timestamp of when the SBOM data was generated.

2.3.2: Is a repository that tracks the use of third-party components in systems implemented and maintained?

To successfully manage TPCs, organizations must establish a data repository for managing appropriate and useful information. A *data repository* is a structure used to collect, manage, and store data. It often refers to a specific setup, such as a group of databases or a data warehouse. Data repositories are important because they support information analysis, information sharing, and reporting activities.

A data repository can help track the use of TPCs and the associated BOMs/SBOMs and provenance data for the system being acquired and developed. Engineers and developers can review the vulnerability and patching history of a system's TPCs and assess the associated security/resilience risks to the system and its operational mission.

A TPC data repository is a valuable tool for managing security/resilience risks in a system. At the same time, using such a repository introduces risks that must be managed. For example, a TPC repository can contain sensitive information about a system that must be protected from unauthorized access.

2.3.3: Are third-party components that are used in the system identified and documented to create a bill of materials/software bill of materials?

A BOM lists a system's hardware, software, firmware, and service components, similar to ingredient information found on food packaging. BOMs also provide information about these components' sources/providers. Information in a BOM can be linked to supporting information about risks, including known vulnerabilities, available updates, and past remediation activities. To ensure that TPC data is current, BOMs should be reviewed and updated regularly over the lifecycle of each TPC.

An SBOM focuses on a system's software and streamlines the management of software components, including tracking licenses and vulnerabilities. It also provides an inventory of a system's software dependencies. Collecting and mapping software dependency information can require manual effort, relying on individual development teams to supply dependency information from system build data, references in source code, and system architecture documentation. Automation can facilitate collecting and managing SBOM data, and while many automated solutions and tools exist for SBOMs, there is no universal solution for every scenario.

2.3.4: Is the provenance of software components (including origin, development history, and change history) established and tracked?

Provenance is the chronology of the origin, development, ownership, location, and changes to a system or system component and its associated data. It may also include the personnel and processes that interact with or make modifications to the system, component, or associated data [NIST 2020].

Software provenance refers to the verifiable information related to software, including its origin, development, and distribution. Software provenance information enables program personnel to verify the authenticity and integrity of software (e.g., developers are who they claim to be, software has not been tampered with after release). This information is also important for understanding the development history of software, establishing how the software has evolved (e.g., added features, security updates), and tracking software artifacts across the systems lifecycle. From a security/resilience perspective, software provenance information enables program personnel to track software licenses, defects, vulnerabilities, and dependencies.

2.3.5: Are third-party component providers (e.g., contractors, suppliers) evaluated and selected based on their use of secure/resilient development practices?

TPC providers develop custom and off-the-shelf components. Since engineers and developers integrate these components into the system being acquired and developed, the system inherits security/resilience risks from them. Therefore, it is important to ensure that TPC providers use appropriate security/resilience development practices that meet or exceed the requirements established by the acquisition program.

Since development practices continually evolve, TPC providers must regularly consult industry standards and maintain current and robust security/resilience management practices so they can effectively manage vulnerabilities, threats, and risks. TPC providers that follow these practices are better able to manage security/resilience risks in their components and communicate relevant risk information to their customers (i.e., the acquisition program).

2.3.6: Is each third-party component's operational risk assessed?

Assessments are an effective way to identify TPC risks and establish mitigation priorities. Risk assessments should be ongoing over the life of TPCs. Assessing TPC risk requires the following:

- defining security/resilience requirements for the system
- establishing a process for conducting the assessment of the system's TPCs, including how to interpret and track the results

To prepare for the assessment, the organization should establish a list of its TPCs. It should then assess each one to gauge the operational risks[30] that it might introduce. A logical starting point is to identify each TPC's known vulnerabilities and assess their potential impact on the system and its mission.

When possible, the risk assessment should incorporate analyses that identify additional security/resilience issues or impending problems from using a TPC, including these factors:

- evaluating the maturity and development practices of the TPC provider
- establishing the stability of the TPC over time
- estimating whether a TPC will reach end of life within the expected lifetime of the system being developed

2.3.7: Is each third-party component monitored for vulnerabilities and available updates?

The acquisition program must establish a process for monitoring and communicating TPC vulnerabilities to engineers and developers as well as other relevant program/system stakeholders. They can use tracking systems to manage information about TPCs and address the risks their vulnerabilities pose.

TPCs should be monitored for changes in their risk profiles, including newly discovered vulnerabilities or changes in end-of-life status as declared by the TPC provider. Monitoring should leverage public resources, such as public vulnerability databases (e.g., the National Institute of Standards and Technology [NIST] Common Vulnerabilities and Exposures [CVE] database) and supplier-specific notification mechanisms (e.g., GitHub, supplier announcement mailing lists).

Programs should consider the following as part of their TPC monitoring and communication process:

- proactively identify new/unknown vulnerabilities before they can be exploited
- complement tracking and management with additional research to ensure that vulnerability information is current and actionable
- incorporate the specialized care required for vulnerabilities that may aggregate to classified levels

2.3.8: Are third-party components prioritized for mitigation based on operational risk?

An acquisition program's approach to risk assessment should incorporate practices for identifying, prioritizing, tracking, and mitigating TPC risks. *Risk prioritization* uses clear criteria to decide which risks are most severe and must be handled first. This prioritization helps the organization manage TPC-related security/resilience risks effectively.

---

[30] *Operational risk* is the risk of loss resulting from inadequate or failed internal processes, policies, systems, or events that disrupt business operations.

Mitigation priorities (e.g., applying software patches) should be established based on these prioritized risks. Since personnel and funding allocated to vulnerability management activities are often limited, a primary goal of risk prioritization is to establish a basis for effectively allocating a program's limited mitigation resources.

Assessing each TPC's known vulnerabilities helps identify risks to the system and its operational mission. Risks introduced by unpatched vulnerabilities should be tracked as part of the acquisition program's risk management activities.

Goal 2.4: Implementation

Security/resilience controls are implemented, and weaknesses and vulnerabilities in software code are assessed and managed.

The purpose of this goal is to build security/resilience into the system and system components by implementing controls and managing weaknesses and vulnerabilities in the code base.

Goal Summary

During system implementation, engineers construct system components that meet stakeholder and system requirements. Implementation activities address security/resilience in two ways. First, security/resilience controls are implemented in the system and system components as specified in the system and software architectures. Second, weaknesses and vulnerabilities in software code are analyzed and addressed.

Security/resilience processes and tools are used to develop the source code for the system's software components and address weaknesses and vulnerabilities. Secure coding practices, code reviews, and code analysis through white-box and black-box testing are important means for identifying and managing vulnerabilities and risks in the code base. Once identified, the coding weaknesses and vulnerabilities must be remediated and tracked, including their associated risk, priority level, and mitigation status.

These implementation activities form this goal's practice questions, which are summarized in the list below and discussed in detail in the *Guidance for Practice Questions* section on page 78.

List of Practice Questions

2.4.1: Are security/resilience controls implemented in the system and system components?

2.4.2: Is an appropriate suite of security/resilience tools integrated into the software development environment?

2.4.3: Are secure coding standards applied?

2.4.4: Are code reviews (e.g., peer reviews) performed to identify weaknesses and vulnerabilities?

2.4.5: 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?

2.4.6: 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?

2.4.7: Are coding weaknesses and vulnerabilities remediated and tracked to resolution?

Context

During system implementation, engineers construct system components that meet stakeholder and system requirements. To be fully effective, implementation activities must ensure that all system components meet these requirements and are designed to adapt to evolving threats and vulnerabilities over the systems lifecycle.

Implementation activities address security/resilience in two ways:

- **Security/resilience controls are employed in the system and system components.** As engineers develop system components, they must implement the security/resilience controls as specified in the system and software architectures.
- **Weaknesses and vulnerabilities**[31] **in software are analyzed and addressed.** Coding defects are a major source of security vulnerabilities and risks. As a result, engineers must identify and remediate software vulnerabilities and their associated risks across the lifecycle. During software code development and software integration, software developers are responsible for minimizing weaknesses and vulnerabilities in the code they develop.

Once system components are constructed, they must be combined to form the system.[32] *Software integration* refers to combining software code from multiple sources (including custom-developed code and off-the-shelf code) into one system, subsystem, or component. The scope of software integration can range from combining relatively small pieces of code into an integrated component to combining several large software components into a large and complex system. Programmers may spend considerable effort on software-integration activities, depending on the size and scale of the development effort, the skills of the programmers, and the lifecycle that is being followed.

Given the complexities involved in implementation and integration, it is understandable that no software is free of risks; defects exist even in the highest quality software. For example, best-in-class code can have up to 600 defects per million lines of code (MLOC), while average-quality code has around 6,000 defects per MLOC, and some of these defects are weaknesses that can lead to vulnerabilities. Research indicates that up to 5% of software defects are security vulnerabilities [Woody 2014]. As a result, best-in-class code can have up to 30 vulnerabilities per MLOC. For average-quality code, the number of security vulnerabilities can be as high as 300 vulnerabilities per MLOC. These insights highlight the importance of reducing security vulnerabilities in code during software development. Secure coding practices, code reviews, and code analysis tools are important ways that known weaknesses and vulnerabilities in code can be identified and corrected.

That said, although security vulnerabilities are reduced during software development, software's development and integration is a continuous process. For most systems, software is deployed in multiple releases or increments, each introducing new capabilities, features, and planned changes. Furthermore, software releases do not stop when a system is deployed; instead, new capabilities, features, and updates are introduced periodically throughout operations and sustainment (O&S).

---

[31] A *weakness* is a condition in a hardware, software, firmware, or service component that, under certain circumstances, could contribute to the introduction of vulnerabilities [MITRE 2023]. A vulnerability is a defect in a software, firmware, hardware, or service component resulting from a weakness that can be exploited to produce a negative impact to the confidentiality, integrity, or availability of system components and their associated data [MITRE 2023]. Therefore, weaknesses are errors or defects that can lead to vulnerabilities. Examples of software weaknesses include buffer overflows and code evaluation and injection.

[32] A software bill of materials (SBOM), which lists the software included in a system or component, provides valuable information that helps programmers manage vulnerabilities and risks during integration.

This continuous evolution requires that software weaknesses and vulnerabilities be continuously managed across the systems lifecycle. Regularly implementing secure coding practices, conducting code reviews, and analyzing code are essential practices for effectively managing security/resilience risks throughout the systems lifecycle.

Competencies

Knowledge Areas	Skills
- computer programming principles - cybersecurity threats and vulnerabilities - software development lifecycle model being used - systems engineering process - security/resilience software engineering principles - controls for protecting systems and information - secure coding principles and concepts - methods for analyzing code, including peer reviews, white-box testing, and black-box testing	- establishing the security/resilience controls for the system and system components - assessing security/resilience controls to determine if they are implemented correctly, operating as intended, and producing desired outcomes - developing secure software according to secure software development methodologies, tools, and practices - writing code in a supported programming language (e.g., Java, C++) - performing code reviews (e.g., peer reviews) - applying white-box testing (e.g., static code analysis) methods, tools, and techniques to identify vulnerabilities - applying black-box testing (e.g., dynamic code analysis, vulnerability scanning) methods, tools, and techniques to identify vulnerabilities - conducting vulnerability scans and recognizing vulnerabilities in system components - remediating or correcting identified coding weaknesses and vulnerabilities

Guidance for Practice Questions

2.4.1: Are security/resilience controls implemented in the system and system components?

As engineers develop system components, they implement the security/resilience controls that are specified in the system and software architectures. Firewalls, intrusion detection systems (IDSs), encryption, and identification and authentication mechanisms are examples of security/resilience controls. Engineers should use best practices when implementing controls, including systems security/resilience engineering methodologies, concepts, and principles. They can also use the results of risk assessments to guide their decisions about the costs, benefits, and tradeoffs associated with different technologies used to implement controls [NIST/DOC 2018].

When implementing security/resilience controls, engineers must also address the quality of the design, development, and implementation of the controls. They must assess controls during system development activities to determine whether the controls are implemented correctly, operating as intended, and producing desired outcomes [NIST/DOC 2018]. Defects and flaws in implemented controls are weaknesses when they can be exploited. It is therefore important to address

implementation defects and flaws during system development activities to enable engineers to apply cost-effective corrective actions [NIST/DOC 2018].

2.4.2: Is an appropriate suite of security/resilience tools integrated into the software development environment?

A development environment is a workspace with a collection of processes and tools used to develop the source code for software components. It supports the end-to-end software engineering process—including development, staging, and production—and automates or facilitates processes for creating, testing, debugging, patching, updating, and maintaining software. The suite of security/resilience tools integrated into the software development environment can include static code analysis, dynamic code analysis, interactive testing, and software composition analysis.

Development teams have a range of options for setting up a development environment; the approach they select can enhance development activities or introduce barriers. For example, it can take significant time and effort to deploy and integrate complex tools into existing development software. Therefore, developers must have the appropriate knowledge, skills, and experience to set up and use the development environment and its processes and tools effectively.

2.4.3: Are secure coding standards applied?

Security is an important factor when selecting a programming language. For example, languages like Rust, known for their memory-safe features, are widely recognized to be more secure. Once a programming language is selected, development teams can implement *secure coding standards*: language-specific rules and guidelines (e.g., for C, C++, Java, Perl) that reduce security weaknesses and errors during development.

These standards, developed through a broad-based effort by members of the software development and software security communities, include guidance for adopting naming conventions, eliminating buffer overflows, and bounds checking variables. When used effectively, these standards prevent, detect, and eliminate errors that could compromise software security.

To effectively integrate secure coding standards into a program's development process, it is important to foster a strong security culture across development teams. This integration of secure coding practices and security/resilience tools into existing workflows ensures that developers apply these standards consistently throughout the development lifecycle.

2.4.4: Are code reviews (e.g., peer reviews) performed to identify weaknesses and vulnerabilities?

A code review is a disciplined engineering practice used for detecting and correcting defects in software code. During a code review, programmers systematically check each other's code for errors, irregular formatting, or inconsistencies with system requirements that may lead to problems during software integration.

Code reviews generally examine four key areas of software weakness:

- defects in the code
- consistency with the system's software requirements and design
- quality of documentation
- consistency with coding standards

Code reviews contribute to the quality control in software development by enabling teams to review their development artifacts early and often. The effectiveness of code reviews in identifying weaknesses and vulnerabilities depends on participants' knowledge, skills, and experience in the selected programming language.

2.4.5: 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?

White-box testing (also known as clear-box testing, glass-box testing, transparent-box testing, and structural testing) is a method of testing that examines the software's internal structure, design, and coding. It is used to verify input/output flows and improve design, usability, and security. White-box testing can be conducted at system, integration, and unit levels of software development and is performed in two parts:

- reviewing the source code to understand its design
- developing and executing test cases

White-box testing can be automated to improve its efficiency, accuracy, and cost-effectiveness; tools used to automate white-box testing must be kept current and maintained.

Static code analysis, or static application security testing (SAST), is a form of white-box testing that identifies weaknesses in static (i.e., non-running) source code. Using SAST, the tester can evaluate a range of static inputs and software source code to detect security weaknesses, performance issues, non-compliance with standards, and outdated programming constructs.

2.4.6: 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?

Black-box testing (also known as functional testing, behavioral testing, and closed-box testing) is a method of testing that examines the functionality of software without knowing its internal code structure, implementation details, and internal paths. During black-box testing, testers analyze the software's requirements and specifications, develop and execute test cases, remediate identified defects, and retest the software to confirm that defects have been corrected. Common types of black-box testing include functional testing, nonfunctional testing, and regression testing.

Black-box testing tools can automate testing and organize test results; these tools must be kept current and maintained.

Dynamic code analysis, or dynamic application security testing (DAST), is a form of black-box testing that identifies potentially exploitable vulnerabilities in the binary code of running software. Using DAST, the tester is not required to know about the technologies or frameworks that underpin the software. Dynamic analysis tools identify both compile-time and runtime vulnerabilities, such as configuration errors that appear only within a realistic execution environment. Since DAST dynamically analyzes software, it can identify runtime vulnerabilities.

2.4.7: Are coding weaknesses and vulnerabilities remediated and tracked to resolution?

Vulnerability remediation involves eliminating identified weaknesses (i.e., defects that can potentially be exploited) and vulnerabilities (i.e., defects that have known exploits). Identified weaknesses and vulnerabilities must be prioritized based on their risk to the mission. Assessing the risk associated with a weakness or vulnerability involves considering several factors, including severity, ease of exploit, exploit availability, available corrective actions, and impact on the mission.

It is crucial to track the status of each identified weakness and vulnerability over time, including the associated risk, priority level, and mitigation status. A tracking system can generate reports for stakeholders about the remediation status of identified weaknesses and vulnerabilities.

Goal 2.5: Test and Evaluation

Security/resilience risks that can affect the integrated system are identified and remediated during test and evaluation.

The purpose of this goal is to verify the system's security/resilience requirements and assess the security/resilience of the system under relevant operational conditions.

Goal Summary

Test and evaluation (T&E) is a process used to examine a system or system components in relation to their requirements and specifications. From a security/resilience perspective, T&E verifies a system's functional and nonfunctional security/resilience requirements and assesses the security/resilience of a system under relevant operational conditions. The goal when assessing security/resilience during T&E is to identity and mitigate *exploitable vulnerabilities* before the system is deployed [DoD 2020b].

Before conducting T&E security/resilience assessments, the program must ensure that required authorization is obtained, test plans and artifacts are prepared, and the activities can be performed in an operationally relevant environment.

Assessing security/resilience in T&E comprises two core activities: vulnerability assessments and adversarial assessments. Vulnerabilities identified during these assessments are analyzed to determine their risk to the system and its mission. Priorities are then established based on the evaluated risks to guide mitigation activities. T&E reports document test results in accordance with the program's test plans and are communicated to stakeholders.

These T&E activities form this goal's practice questions, which are summarized in the list below and discussed in detail in the *Guidance for Practice Questions* section on page 84.

List of Practice Questions

2.5.1: Is there a requirement to obtain authorization to assess security/resilience during test and evaluation?

2.5.2: Are test plans and artifacts for security/resilience developed and updated?

2.5.3: Are security/resilience test-and-evaluation activities performed in an operationally relevant environment?

2.5.4: Are tests of the system and software security/resilience requirements performed?

2.5.5: Are vulnerability assessments of the system performed?

2.5.6: Are adversarial assessments (e.g., red team exercises) of the system performed?

2.5.7: Are security/resilience risks identified by analyzing vulnerabilities discovered during test and evaluation?

2.5.8: Are security/resilience risks identified during test and evaluation analyzed?

2.5.9: Are security/resilience risks identified during test and evaluation mitigated and tracked?

2.5.10: Are security/resilience risks identified during test and evaluation communicated to stakeholders?

Context

The purpose of T&E is to verify security/resilience requirements, identify problems, characterize system capabilities and limitations, and manage technical and program risks. T&E is conducted across the systems lifecycle to assist in design and development activities and to verify that technical performance requirements are met. As such, it is an essential aspect of a program's effort to ensure adequate security/resilience in a system or system component.

From a security/resilience perspective, T&E verifies a system's security/resilience requirements and assesses the security/resilience of a system under relevant operational conditions. Assessing security/resilience during T&E comprises two core activities:

- **Vulnerability assessments** are examinations of a system to establish the effectiveness of its security/resilience controls and identify vulnerabilities.
- **Adversarial assessments** evaluate the system's vulnerabilities using realistic tactics, techniques, and procedures while in a relevant operating environment.

Vulnerabilities identified during T&E are evaluated for the risks they pose to the system and its mission. The risks are used to establish priorities that guide the implementation of mitigations. System testers document test results in accordance with the program's test plans. Test results demonstrate that the test plans were performed as specified. They also indicate which requirements system testers verified and validated and the compliance status of security/resilience controls. Vulnerabilities and their associated risks and mitigations are documented and tracked by system testers.

Programs use T&E results to ensure that adequate development progress is being made across the systems lifecycle. These results also provide insight into the system's capabilities, limitations, and deficiencies. A range of program stakeholders can contribute input to T&E activities to establish testing rigor and ensure that relevant parties are engaged in the process.

For major defense acquisition programs (MDAPs), T&E activities include two distinct events, each of which requires a vulnerability assessment and adversarial assessment:

- **Developmental Test and Evaluation (DT&E)** verifies that the system is built correctly in accordance with the specification and contract.
- **Operational Test and Evaluation (OT&E)** validates that the system can successfully accomplish its mission in a relevant operational environment. OT&E is performed by an organization that is independent of the acquisition program.

Competencies

Knowledge Areas	Skills
- technologies that are being acquired and developed - system and software validation and verification principles and methods - security/resilience risk management principles and methods - cybersecurity threats and vulnerabilities - threat exploitation techniques - vulnerability assessment methods, tools, and techniques - adversarial assessment methods, tools, and techniques	- performing verification of security/resilience requirements - testing to detect misconfigured devices and nonfunctional protections - verifying cybersecurity functionality to ensure that security/resilience controls are working as intended in a mission context - conducting vulnerability scans and recognizing vulnerabilities in security systems - conducting and/or supporting authorized penetration testing on enterprise network assets - executing adversarial exploits to alter, compromise, and corrupt targeted systems in a relevant operating environment - preparing reports that identify technical and procedural findings and recommend remediation strategies/solutions

Guidance for Practice Questions

2.5.1: Is there a requirement to obtain authorization to assess security/resilience during test and evaluation?

When tests are performed in an operationally relevant environment or when live operational data is required to support T&E activities, the acquisition program may need to obtain authorization to conduct the testing. In the Department of Defense (DoD), to perform tests in an operationally relevant environment or with live operational data, acquisition programs must obtain either an interim authorization to test (IATT) or an authorization to operate (ATO).

An IATT is approved by the authorizing official (AO) and provides temporary authorization to test a system in a specified operational environment or with live data for a specified time frame. An ATO is granted when the AO determines that assigned cybersecurity controls have been implemented adequately, the system's residual risk is acceptable, and operational testing can begin. Either authorization means that assigned security/resilience controls have been implemented adequately and the system's residual operational/mission risk is acceptable enough to allow T&E activities to commence.

2.5.2: Are test plans and artifacts for security/resilience developed and updated?

Test plans establish specific and detailed testing approaches and procedures to be implemented. The level of detail in a test plan should reflect the risk and complexity of the system or system component being tested. Key aspects of a test plan include appropriate, validated test objectives and a means of measuring how well those objectives are met. Key testing artifacts should be established and included in a test plan. Test plans should be provided to program/system stakeholders as part of the program's communication and coordination activities.

An acquisition program's planned T&E activities are typically documented in the T&E plan that describes the overall structure and objectives of the T&E activities and identifies the resources necessary to accomplish each activity.[33] It provides a framework for developing detailed test plans, and it documents the T&E schedule and resources.

2.5.3: Are security/resilience test-and-evaluation activities performed in an operationally relevant environment?

An operationally relevant environment includes the set of operational conditions—selected by system users in coordination with the testing organization—that represent the range of a system's operations. Operationally relevant environments include a broad range of options, including models, simulations, testbeds, prototypes, full-scale engineering development models of the system, and the actual environment where the system is deployed.

The challenge is to cost-effectively provide an operationally relevant environment that does not impact production or operations. Testing a system in its operational environment can be done when production volume or risk to operations is low (e.g., during test windows). Alternatively, specialized environments (e.g., models, simulations, testbeds, prototypes) can be built separately from the operational environment specifically for testing. In general, using a combination of specialized test environments and testing a system in its operational environment helps increase testing quality and manages disruption risk.

2.5.4: Are tests of the system and software security/resilience requirements performed?

Requirements-based testing helps identify issues related to performance, reliability, scalability, usability, and security/resilience. This testing approach must satisfy two key objectives:

1. Validate that requirements are correct, complete, unambiguous, and logically consistent.
2. Include a set of test cases that verify whether the system meets its requirements.

Rigorous T&E activities include developing and executing tests cases for the system's security/resilience requirements. These test cases enable program/system stakeholders to verify that the system will satisfy its security/resilience requirements. Any defects or issues discovered during testing are monitored, logged, and reported.

Testers collect information about the defects and their impacts on security/resilience requirements. The program should have a procedure that engineers and developers can use to resolve defects that are reported and verify solutions through retesting.

[33] A DoD acquisition program's planned testing activities are documented in a Test and Evaluation Master Plan (TEMP), which is the overarching document for managing a program's T&E activities.

2.5.5: Are vulnerability assessments of the system performed?

Assessing and managing vulnerabilities is a leading practice for acquisition programs. In vulnerability assessments, an information system or product is examined to determine the adequacy of security measures, identify deficiencies, provide data to help predict the effectiveness of proposed controls, and confirm the adequacy of controls after their implementation [NIST/DOC 2020]. Vulnerability assessments often start during system development (e.g., during coding and implementation activities) and continue throughout the systems lifecycle.

Vulnerabilities are typically detected using automated tools that scan a system and produce a report. A vulnerability assessment report provides a list of the vulnerabilities found, their estimated severity levels, and recommendations for remediating them. Vulnerabilities identified during the assessment should be evaluated for the risks they pose to the system and its mission. These assessed risks are used to establish risk mitigation priorities. (See Practices 2.5.7 through 2.5.9 for more information about risk management activities performed during T&E.)

2.5.6: Are adversarial assessments (e.g., red team exercises) of the system performed?

In adversarial assessments, the system's security/resilience is evaluated using realistic tactics, techniques, and procedures while in an operationally relevant environment. These assessments are performed on systems or system components during T&E after vulnerability assessment activities are complete.

To conduct an adversarial assessment, trained experts perform planned and coordinated simulated attacks on a system or technology environment. They use a variety of tools and techniques to identify vulnerabilities and facilitate the ongoing management of risk. These assessments evaluate the system's ability to achieve its mission while the system is subjected to validated and realistic cyber threats.

Vulnerabilities identified during adversarial assessments should be evaluated for the risks they pose to the system and its mission. These assessed risks are used to establish risk mitigation priorities. Organizations often perform regular adversarial assessments because of attackers' ongoing efforts to use new methods to exploit security vulnerabilities and weaknesses. (See Practices 2.5.7 through 2.5.9 for more information about risk management activities performed during T&E.)

2.5.7: Are security/resilience risks identified by analyzing vulnerabilities discovered during test and evaluation?

Vulnerability assessments and adversarial assessments conducted during T&E can produce a long list of vulnerabilities that must be evaluated for the risks they pose to the system and its mission. The risk associated with each vulnerability is established by considering several factors, including severity, ease of exploit, exploit availability, available corrective actions, and impact on the mission. Security/resilience risks are documented in a prescribed format in preparation for risk analysis.

2.5.8: Are security/resilience risks identified during test and evaluation analyzed?

Risk analysis evaluates and prioritizes security/resilience risks, and risk evaluation establishes probability, impact, and risk exposure measures for each identified security/resilience risk. Risk evaluation can be qualitative (e.g., based on low, medium, and high criteria) or quantitative (based on numerical values). In practice, when evaluating security/resilience risks, most organizations use a qualitative approach, primarily due to the lack of objective risk data.

Risk prioritization determines the order used to address risks based on their probability, impact, and risk exposure measures. Clear criteria should be used to evaluate and prioritize risks.

2.5.9: Are security/resilience risks identified during test and evaluation mitigated and tracked?

Risk management methods offer several options for responding to a risk, including the following:

- accept (take no action)
- transfer (shift to a third party)
- avoid (restructure activities to eliminate a risk)
- watch (monitor for changes)
- mitigate (take action to reduce or contain a risk)

Assessors select an option for each risk based on its measures (i.e., probability, impact, and risk exposure) and relative priority, which are established during risk analysis. High-priority security/resilience risks are generally mitigated immediately, and plans are developed to reduce or contain these risks.

Mitigation plans for security/resilience risks incorporate the following basic strategies:

- **Protect.** Reduce vulnerability to threats and minimize any consequences that might occur.
- **Detect.** Identify the occurrence of a security/resilience threat (i.e., cyber attack).
- **Respond.** Take action to counteract a detected threat (i.e., cyber attack) and minimize consequences, losses, and damages.
- **Recover.** Restore access to and functionality of a system (or systems) after a risk's consequences, losses, and damages are realized.
- **Adapt.** Enable a sustained capability to accommodate changes in a system's risk environment, including changes to threats, vulnerabilities, mission, and technologies.

Once a plan is developed, documented, and approved, it is implemented and tracked. Risk tracking monitors mitigation plans for efficiency and effectiveness. Plans should be monitored to ensure that mitigation actions are being performed as specified and that security/resilience controls are implemented correctly, operating as intended, and producing desired outcomes. A tracking system provides reports that can be shared with program/system stakeholders to update them about mitigation plan status.

2.5.10: Are security/resilience risks identified during test and evaluation communicated to stakeholders?

Security/resilience risks cannot be addressed if they are not communicated to and understood by the program's decision makers. Program/system stakeholders can provide valuable information about how to prioritize and manage a system's security/resilience risks. They can also directly or indirectly support risk mitigation activities. Therefore, the program must establish a process for communicating the results of vulnerability and adversarial assessments to all relevant program/system stakeholders and provide the opportunity for engagement.

While the volume of T&E information can be significant, it is useful to keep program/system stakeholders appropriately informed. Ensuring they have access to security/resilience test reports is a leading practice for acquisition programs.

Goal 2.6: Authorization to Operate

The operation of the system is authorized, and the residual risk to operations is explicitly accepted.

The purpose of this goal is to enable the authorizing official (AO) to determine whether to allow the system to operate on the organization's networks based on an analysis of the system's security/resilience controls and risks.

Goal Summary

An *authorization to operate (ATO)* is an official management decision that the AO[34] makes to explicitly accept the risk to organizational operations and authorize the operation of a system. The goal is to ensure that the system's security/resilience risks are within an acceptable tolerance level. The ATO process consists of an assessment and an authorization and is reassessed periodically:

- The **assessment** evaluates the system's controls, risks, and mitigation strategies according to the ATO plan. Information about the architecture is analyzed and documented. A risk assessment is performed, and assessment results and a plan for mitigating risks are documented and communicated to program/system stakeholders.
- The **authorization** involves creating an authorization package that the AO uses to decide whether the risk to the system is acceptable. The AO determines whether to allow the system to operate on the organization's networks and communicates the decision to program/system stakeholders.

These ATO activities form this goal's practice questions, which are summarized in the list below and discussed in detail in the *Guidance for Practice Questions* section on page 92.

List of Practice Questions

2.6.1: Is a plan developed and documented for assessing security/resilience controls and risks?

2.6.2: Is the architecture (including dataflows, interfaces, and hardware/software inventories) documented for the system being submitted for authorization?

2.6.3: Are security/resilience controls assessed and documented in accordance with the assessment plan?

2.6.4: Are remediation actions implemented to correct identified deficiencies in system controls?

2.6.5: Are security/resilience risks (including threats and vulnerabilities) assessed and documented for the system?

2.6.6: Are plans developed and documented for managing security/resilience risks?

2.6.7: Are assessment results formally communicated to program/system stakeholders?

2.6.8: Is an authorization package developed and submitted to the authorizing official for approval?

[34] The AO is the individual with the authority to formally assume responsibility for operating a system at an acceptable level of risk to agency operations, agency assets, or personnel [NIST 2022].

2.6.9: Is an authorization decision made based on an analysis of the system's security/resilience risks?

2.6.10: Is the authorization decision formally documented and communicated to program/system stakeholders?

2.6.11: Is the authorization to operate the system reassessed periodically?

Context

An ATO is part of a broader risk management process that an acquisition program implements in accordance with guidance specified in the National Institute of Standards and Technology (NIST) Risk Management Framework (RMF) as defined in NIST Special Publication 800-37, *Risk Management Framework for Information Systems and Organizations* [NIST/DOC 2018].[35] NIST defines an ATO as

> *the official management decision given by a senior organizational official to authorize operation of an information system and to explicitly accept the risk to organizational operations (including mission, functions, image, or reputation), organizational assets, individuals, other organizations, and the Nation based on the implementation of an agreed-upon set of security controls* [NIST/DOC 2018]

In the context of this discussion, the system undergoing the ATO process is referred to as the *system of interest*.[36] A key aspect of the ATO process is clearly defining the authorization boundary for the system of interest. This boundary identifies all components to be included in assessment and authorization activities, establishing the scope for the ATO process.

The authorization boundary excludes the following two types of support systems from the ATO process:

- **Separately authorized systems** are operational systems that are connected to the system of interest as part of a larger *system of systems (SoS)*.[37] During operations, the system of interest exchanges data and services with other independently managed systems to support a mission. The independently managed systems are outside of the authorization boundary for the

---

[35] National Institute of Standards and Technology Special Publication 800-37, *Risk Management Framework for Information Systems and Organizations* (NIST RMF) provides a comprehensive, flexible, repeatable, and measurable seven-step process that organizations can use to manage security/resilience risks. The Risk Management Framework (RMF) steps are Prepare, Categorize, Select, Implement, Assess, Authorize, and Monitor. The ATO process covers the RMF Assess and Authorize steps. The Categorize, Select, and Implement steps are performed during system acquisition and development. The Monitor step is performed during Operations and Sustainment (O&S). Refer to the report *Risk Management Framework for Information Systems and Organizations* for detailed information about the RMF and its seven-step process [NIST/DOC 2018].

[36] A *system of interest* is defined as the focus of the systems engineering effort [NIST 2021]. It refers to the system that is being acquired and developed and that is undergoing an assessment. A system of interest includes system components, system component interconnections, and the environment where they are placed [NIST 2021].

[37] An *SoS* is defined as a set or arrangement of interdependent systems that are related or connected (i.e., networked) to provide a given capability [Levine 2003].

system of interest [NIST/DOC 2018]. Those systems receive ATOs independent of the system of interest.

- **Enabling systems** support a system as it is being acquired and developed. Examples of these systems include engineering, development, test, and training systems. Enabling systems are not necessarily delivered with the system of interest and do not necessarily exist in the operational environment of the system of interest [NIST 2021]. As a result, enabling systems are typically outside of the authorization boundary [NIST/DOC 2018].

An ATO assessment requires comprehensive evaluations (i.e., a control assessment) of controls and risks for the system of interest. The goal of a control assessment is to ensure that security/resilience requirements are met for the system of interest, and it examines the following two types of controls:

- **System controls** are security/resilience safeguards or countermeasures that are implemented at the system level. This control assessment determines whether system controls are implemented correctly, operating as intended, and producing the desired outcome [NIST/DOC 2018].
- **Common controls** are security/resilience controls that can be inherited by one or more systems across an enterprise. Common controls are identified across the enterprise where the system of interest is deployed. They are allocated to organizational entities designated as *common control providers*. System assessors refer to authorization packages prepared by common control providers when making determinations about the adequacy of common controls that the system of interest inherits.

The ATO process consists of the following two main activities:

- **Assessment.** During this activity, security/resilience risks are identified, and risk management strategies are developed for the system of interest. Assessors identify and characterize threat sources and threat events for the system of interest. Next, vulnerabilities and predisposing conditions are identified to help assessors understand (1) the nature of system vulnerabilities and threats and (2) the degree to which the system is vulnerable to the identified threats. Assessors use these identified threats and vulnerabilities and information about potential mission impacts to identify and prioritize the system's security/resilience risks. Finally, strategies for managing high-priority security/resilience risks are identified and documented to ensure that the system's security/resilience risks are within an acceptable tolerance level.
- **Authorization.** This activity provides organizational accountability by requiring the AO to determine whether the security/resilience risks (including supply chain risks) for the system of interest are acceptable. Based on the results of the ATO assessment, the program develops remediation plans that document how identified risks, issues, and vulnerabilities will be addressed, including the resources required to accomplish each element of the plan. An authorization package is submitted to the AO for an authorization decision. The AO analyzes the information in the package and finalizes the determination of risk. The authorization decision is a formal statement by the AO regarding acceptance of the risk associated with operating a

system and is expressed as an ATO, interim authorization to test (IATT),[38] or denial of ATO (DATO).[39]

Competencies

Knowledge Areas	Skills
- RMF requirements - assessment and authorization process - controls for protecting systems and information - security/resilience risk management - technologies that are being acquired and developed - security/resilience principles used to manage risks related to the use, processing, storage, and transmission of information or data - system and network security/resilience architecture concepts - structured analysis principles and methods	- establishing acceptable risk levels for the system of interest - reviewing authorization and assurance documents to confirm that the level of risk is within acceptable limits for the system under evaluation - establishing the security/resilience controls for the system of interest - evaluating the system and software architectures to determine whether controls and mitigations are adequate - developing authorization packages for the system of interest - managing and approving authorization packages for the system of interest - providing information about assessment and authorization activities to system stakeholders

Guidance for Practice Questions

2.6.1: Is a plan developed and documented for assessing security/resilience controls and risks?

A plan for assessing security/resilience controls and risks is developed by the assessors based on information in program documents, including architecture, control, and risk information. The assessment plan does the following:

- establishes the authorization boundary for the system
- defines objectives for control and risk assessments
- documents assessment procedures

The assessment plan is reviewed and approved by the AO or a designated representative to help ensure that it is consistent with the organization's security/resilience objectives. The approved assessment plan is then provided to stakeholders.

---

[38] An *IATT* is a temporary authorization to test an information system in a specified operational information environment within the time frame and under the conditions or constraints enumerated in the written authorization [CNSS 2022].

[39] A *DATO* indicates that there are major weaknesses or deficiencies in the security controls employed within or inherited by a system.

2.6.2: Is the architecture (including dataflows, interfaces, and hardware/software inventories) documented for the system being submitted for authorization?

System and software architectures are analyzed to identify any security/resilience concerns related to the system's design. An architecture report is then written that provides the following:

- a description of the system being submitted for authorization, including architectural drawings and diagrams; descriptions of dataflows and types, including ports and protocols; external and internal interfaces; hardware and software inventories; and any other system-unique characteristics
- an overview of the system, including its mission and operating environment
- an account of all system security/resilience concerns that were identified

Architecture models can be used to represent some of the system's architecture information, depending on whether the program intends to use digital engineering artifacts produced by model-based engineering methods.

2.6.3: Are security/resilience controls assessed and documented in accordance with the assessment plan?

Security/resilience controls are assessed in accordance with the assessment procedures defined in the assessment plan. These control assessments determine whether controls are implemented correctly, operating as intended, and producing the desired outcome (i.e., meeting security/resilience requirements for the system and the organization).

Assessors prepare a security assessment report (SAR)[40] for the AO. A SAR documents the control assessment findings, including a list of recommended corrective actions for any weaknesses or deficiencies identified in the security/resilience controls. The SAR's detailed information helps determine the effectiveness of the controls implemented within or inherited by the information system.

2.6.4: Are remediation actions implemented to correct identified deficiencies in system controls?

The SAR describes any control deficiencies that remained unresolved during system development or that were discovered after system development. System stakeholders, including the AO and system owner, may determine that some deficiencies represent unacceptable risk and require immediate remediation. They can also decide to correct certain deficiencies that can be remediated quickly and easily with existing resources.

[40] The Committee on National Security Systems (CNSS) notes that a SAR provides a disciplined and structured approach for documenting the assessor's findings and recommendations for correcting any identified vulnerabilities in the security controls [CNSS 2022].

If initial remediation actions are taken, then the affected controls are reassessed. Assessors update the SAR to reflect the findings of the reassessment; however, these updates must not change the original assessment results. Completed remediation actions can also be documented in an initial version of a risk mitigation plan for the system, such as the Plan of Action and Milestones (POA&M).[41]

2.6.5: Are security/resilience risks (including threats and vulnerabilities) assessed and documented for the system?

A risk assessment is performed to assess, identify, and prioritize the system's security/resilience risks. To prepare, assessors identify and document the following aspects of the risk assessment method: purpose, scope, assumptions, constraints, information sources, risk model, and analytic approach.

When conducting the assessment, assessors generate the following risk data: threats, vulnerabilities, predisposing conditions, likelihood of threat occurrence, magnitude of threat impact, risks, and risk management strategies. Risk data generated during the assessment is documented in a risk assessment report (RAR).[42]

2.6.6: Are plans developed and documented for managing security/resilience risks?

To ensure that risks are kept at or below an acceptable tolerance level, assessors identify, evaluate, prioritize, and select strategies for managing the risks. Plans are then developed for mitigating high-priority security/resilience risks to the system. These risk mitigation plans document the actions to be taken, who is responsible for completing them, the timetable for their completion, and the resources required for their completion. Plans developed during the assessment are documented in the RAR; risk mitigation actions that require future implementation are documented in the POA&M.

---

41 A *POA&M* is a document that identifies tasks that need to be accomplished. It details resources required to accomplish the elements of the plan, milestones for meeting the tasks, and the scheduled completion dates for the milestones [NIST/DOC 2020].

42 The CNSS notes that an *RAR* is a report that contains the results of performing a risk assessment or the formal output from the process of assessing risk [CNSS 2022].

2.6.7: Are assessment results formally communicated to program/system stakeholders?

Assessment results are documented in the SAR and RAR. The content and format of these reports are determined by the organizations that are responsible for conducting the control and risk assessments. The assessment results in these reports are communicated to program/system stakeholders to support the program's risk responses.

Assessors can communicate these results to program/system stakeholders through various channels, such as executive briefings, written communications, and dashboards. The methods of communicating control and risk information across the program can be formal or informal. Guidance about risk communication and reporting requirements is developed during risk assessment preparation and may be included in the program's risk management plan.

2.6.8: Is an authorization package developed and submitted to the authorizing official for approval?

An authorization package is assembled and submitted to the AO for an authorization decision for the system of interest. The authorization package includes the SAR, RAR, and POA&M. The AO determines the authorization package's format and identifies/requests additional supporting information, artifacts, and references that should be included. The package includes the information the AO needs to make a risk-based decision about whether to authorize the operation of the system.

The program maintains version and change control of the authorization package to help manage changes and updates to plans, assessment reports, and the POA&M. These activities facilitate the program's ongoing risk management and authorization activities and provide useful input for any necessary reauthorization activities.

2.6.9: Is an authorization decision made based on an analysis of the system's security/resilience risks?

The AO analyzes the information in the authorization package and finalizes the determination of risk. The authorization decision is the AO's formal statement regarding the acceptance of the risk associated with operating the system of interest. The explicit acceptance of risk is the AO's sole responsibility and cannot be delegated. This decision is expressed as one of the following: an ATO, IATT, or DATO.

The AO considers many factors when deciding whether the risk to the organization's operations is acceptable. To help the AO develop a thorough understanding of the system's security/resilience risks, the AO meets with the assessors, the system owner, and other program/system stakeholders, as needed, to answer questions and clarify issues.

Balancing security/resilience considerations with mission and business needs is paramount to achieving an acceptable risk-based authorization decision.

2.6.10: Is the authorization decision formally documented and communicated to program/system stakeholders?

The ATO is a record of the AO's final security/resilience authorization decision. It includes the following information:

- authorization date
- authorization termination date
- overall residual risk
- any limitations or restrictions placed on the operation of the system of interest

The signed ATO is then provided to the program manager and other program/system stakeholders as appropriate.

2.6.11: Is the authorization to operate the system reassessed periodically?

The AO establishes the authorization termination date, which indicates when the ATO expires. An authorization can be valid for up to three years, depending on whether significant changes occur that affect the potential risk level of operating the system of interest. Reaching the termination date for a system or changes to a system's operating security/resilience risk level can trigger the assessment and authorization activities of the ATO process.

Some organizations are beginning to use a continuous authorization to operate (cATO) process instead of the traditional ATO process. A cATO authorizes the platform, process, and the team that developed the system of interest under a continuous monitoring process that maintains the residual risk within the risk tolerance of the AO.

To achieve a cATO, the AO must demonstrate three main competencies [DoD 2022d]:

- ongoing visibility of key cybersecurity activities inside the system boundary with robust continuous monitoring of RMF controls
- the ability to conduct an active cyber defense to respond to cyber threats in real time
- the adoption and use of an approved DevSecOps[43] reference design[44]

A cATO requires that the system's owner coordinate with service providers to continuously monitor and assess security/resilience controls within the system's security baseline. They must demonstrate their ability to integrate the automation and continuous monitoring prior to entering into a cATO status [DoD 2022d]. A cATO does not expire as long as the required real-time risk posture is maintained. (Not every system qualifies for a cATO.)

---

[43] DevSecOps stands for Development, Security, and Operations.

[44] A *reference design* is a technical blueprint of a system that is intended for others to copy. It defines the essential elements of the system. Organizations using a reference design can enhance or modify the design as needed to meet organization-specific requirements. The *DoD Enterprise DevSecOps Reference Design* leverages a set of hardened DevSecOps tools and deployment templates that enable DevSecOps teams to select the appropriate template for the program application capability to be developed [DoD 2019, DoD 2010].

Goal 2.7: Deployment

Security/resilience is addressed in transition and deployment activities.

The purpose of this goal is to ensure that security/resilience is considered during all transition and deployment activities.

Goal Summary

Deployment focuses on transitioning (1) system capabilities to end users and (2) support and maintenance responsibilities to the post-deployment support organization. During system deployment, security/resilience must be addressed—including planning; training development and delivery; document preparation; support tool development; and system/component transport, installation, and verification. The confidentiality and integrity risks to the system's sensitive data must also be protected during deployment.

These deployment activities form this goal's practice questions, which are summarized in the list below and discussed in detail in the *Guidance for Practice Questions* section on page 98.

List of Practice Questions

2.7.1: Is a plan for transitioning the system (or system components) into operations and sustainment developed and agreed to by relevant stakeholders?

2.7.2: Are security/resilience training, documentation, and support tools for the system provided to operators/maintainers and users?

2.7.3: Is responsibility for managing security/resilience risks after deployment transferred to the operational support organization?

2.7.4: Are system components protected from tampering and modification during their transport and installation?

2.7.5: Is the integrity of all deployed system components verified?

2.7.6: Are confidentiality and integrity risks for sensitive data (e.g., passwords, tokens) mitigated adequately for software that operates in the operational environment?

Context

Deployment is the process of installing, testing, and implementing a system and its components in an operational environment. A successful system deployment can improve the efficiency, productivity, and satisfaction of users while minimizing the risks, costs, and disruptions associated with the change.

System deployment typically involves transitioning system capabilities to end users and support and maintenance responsibilities to the post-deployment operations and sustainment (O&S) organization [SEBoK 2023]. It often involves conducting demonstration tests and gradually phasing out legacy systems that are being replaced. Plans and clear criteria for these activities must be developed and approved by relevant program/system stakeholders.

Competencies

Knowledge Areas	Skills
- technologies being acquired, developed, operated, and sustained - system requirements and architecture - technology transition models and techniques - training and education models and techniques - document preparation - software engineering and development - security/resilience risk management	- developing a plan for transitioning a system from development to operations - developing training materials - presenting training materials - developing operator/maintainer and user manuals - developing support tools for operators/maintainers - providing help desk support for the system - providing consulting support for the system - evaluating and managing system transport and installation risks

Guidance for Practice Questions

2.7.1: Is a plan for transitioning the system (or system components) into operations and sustainment developed and agreed to by relevant stakeholders?

Developing and documenting a plan for deploying a system and its components into O&S is an important acquisition and development activity. A deployment plan documents the following:

- activities needed to manage the transition of the system to its intended operational environment
- key roles and responsibilities for internal and external stakeholders, including suppliers
- mitigation plans for high-priority transition risks
- descriptions of the planned evolution of the system and its capabilities
- approaches for managing evolving threats/risks
- a rollback plan to follow if deployment fails
- eventual removal of the system from operational use (i.e., decommissioning and disposal)
- all security/resilience activities that must be addressed during transition and deployment
- risks to a system's deployment and the associated mitigation actions
- basic procedures for continuously managing security/resilience during O&S, establishing a roadmap for a successful system transition

A deployment plan helps establish a common understanding of activities, roles, responsibilities, and schedule. Therefore, when developing one, planners should engage acquisition, engineering, and O&S stakeholders to draw on their expertise, gain their support for system transition and deployment activities, and ensure that roles and responsibilities for the system are firmly established throughout its transition to operations.

2.7.2: Are security/resilience training, documentation, and support tools for the system provided to operators/maintainers and users?

Artifacts such as training, documentation, and support tools can help facilitate transition of a system to operators/maintainers and users:

- **Training.** The foundation for successful transition and deployment is training. An effective training program helps ensure that individuals can effectively carry out their security/resilience roles with the necessary proficiency and readiness. The program is responsible for developing and verifying training materials, which can be delivered in person or by computer. Security/resilience training should be provided for a variety of audiences, including operators/maintainers, users, managers, and contractors.
- **Documentation.** The program is also responsible for creating and delivering system documentation, which can be provided in digital or hard-copy formats. System documentation should include guidance for operating, maintaining, and using the system in a secure/resilient manner. Documentation helps provide clarity about technical and risk management details and forms the basis for consistent and predictable system support.
- **Support tools.** Operators/maintainers use tools to support security/resilience practices for a system, including monitoring, access management, and configuration management. O&S personnel must be prepared to maintain the system's tool suite as operations moves to a steady-state support model. The program may therefore (1) develop or procure tools that operators/maintainers use to ensure that the system functions as intended and is reliable, secure, and resilient and (2) provide them to the system's O&S organization.

Those responsible for developing and delivering documentation, support tools, and training must be familiar with the operational environment. Understanding the perspectives of operators/maintainers and users enables developers to provide documentation, support tools, and training that address the needs of the target audience. To ensure an effective transition from development to operations, information must be tailored to the operational audience.

Transitioning these artifacts—training, documentation, and support tools—to an O&S organization requires careful management and collaboration among acquisition, engineering, and O&S stakeholders to ensure that the transition is successful, effective, and efficient. Since these artifacts can help facilitate transitioning a system to operators/maintainers and users, they should be reviewed and updated when new system capabilities are released.

Even though providing these artifacts is critical to the system's successful transition, providing them to the O&S organization may not be sufficient to ensure that a system is operated in a secure/resilient manner. System operators/maintainers and users often need additional support. To facilitate system transition, the acquisition program and O&S organization should encourage personnel to take advantage of information-sharing opportunities, help-desk support, and consultants, as needed.

2.7.3: Is responsibility for managing security/resilience risks after deployment transferred to the operational support organization?

Deploying a system into its operational environment requires transferring its management, including risk management, from the acquisition program to an O&S organization. Establishing clear responsibility for the security/resilience risks of the system being deployed into the operational environment is crucial, and the handoff process must be managed carefully. Since transitions pose a variety of challenges, it is essential that all involved personnel thoroughly understand their designated roles and the schedule and timing of transition activities.

Handoffs are more effective when they include personnel who understand both environments and can bridge communication and terminology gaps introduced by the differences between the environments.

2.7.4: Are system components protected from tampering and modification during their transport and installation?

During system transition, there is a higher likelihood that tampering and modification can occur due to (1) the increased number of personnel involved and (2) the frequent interactions among acquisition, development, and O&S personnel. Ensuring a high level of security/resilience vigilance helps mitigate risks and challenges as a system is transitioned from acquisition and development to O&S. Therefore, a system and its components must be protected from adversaries and across all lifecycle activities, including deployment.

Protection mechanisms during system transport and installation include the following:

- **Physical controls** include secure packaging with tamper-evident seals or locking mechanisms, controlled chain of custody, controlled physical access, and post-installation verification.
- **Technical controls** include digital signatures, code integrity checks during installation, hash value verification, encryption, code obfuscation, and vulnerability scanning.

2.7.5: Is the integrity of all deployed system components verified?

Maintaining system security/resilience during the system's transition to operations requires having an accurate inventory of components and taking the appropriate steps to verify their integrity. To take these steps, acquisition, development, and O&S personnel must first (1) understand how security/resilience risks can affect system deployment and (2) work to mitigate those risks. Likewise, personnel involved in deployment must verify the integrity of system components that are being deployed into the operational environment.

Establishing controls across the systems lifecycle can increase confidence in the integrity of all deployed system components. Examples of controls that can be used to verify the integrity of deployed system components include the following:

- hash value verification
- digital signatures
- code signing
- file integrity monitoring
- configuration management
- version control

Automation can also be used to (1) streamline the deployment process, (2) help ensure the integrity of system components, and (3) provide a consistent and repeatable approach for deploying system components.

2.7.6: Are confidentiality and integrity risks for sensitive data (e.g., passwords, tokens) mitigated adequately for software that operates in the operational environment?

Acquisition, development, and O&S personnel are involved in deployment activities, which potentially provide many personnel with access to sensitive system data (e.g., passwords, tokens). Confidentiality and integrity risks associated with this data must be mitigated adequately for software that operates in operational environments; access to this data must be managed carefully, especially during system deployment.

Maintaining secure access to sensitive system data requires periodic and systematic reviews of credentials to verify that access is granted only to authorized personnel or systems. Effectively managing administrative access rights during deployment is crucial since passwords, cryptographic keys, and authorization tokens are often required for software to operate. The security/resilience of the system depends on maintaining the confidentiality of this sensitive data.

Sensitive system data should not be checked into version control systems or embedded in source code. Whenever possible, sensitive system data should be stored in an appropriate secret management system or encrypted with a key management system. Access to sensitive system data must always be strictly limited, and access to it must only be granted only when necessary.

Goal 2.8: Operations and Sustainment

Security/resilience risks and issues are identified and resolved as the system is used and supported in the operational environment.

The purpose of this goal is to assess and manage security/resilience risks and issues periodically as the system is being used and supported.

Goal Summary

From a security/resilience perspective, an operations and sustainment (O&S) organization must implement a risk management capability to effectively manage security/resilience risks to the operational system.[45] Managing security/resilience risks during O&S requires implementing a baseline configuration for the operational system, performing periodic risk and vulnerability assessments, and implementing corrective actions as appropriate.

Assessments are normally performed periodically and provide a snapshot of the system's risks. System and network monitoring provides system operators/maintainers with information about changes to a system's risk profile that occur between risk assessments. Monitoring activities focus on detecting cyber threats and data breaches in systems and networks. Effective monitoring helps an O&S organization detect cyber attacks early and respond to them before they cause damage and disruption.

Finally, O&S activities require a system to be decommissioned when it has reached the end of its useful life. A system must be disposed of in accordance with legal and regulatory requirements and policies related to safety, security/resilience, and environmental considerations.

These O&S activities form this goal's practice questions, which are summarized in the list below and discussed in detail in the *Guidance for Practice Questions* section on page 105.

List of Practice Questions

2.8.1: Is a baseline security/resilience configuration for the system defined and implemented?

2.8.2: Are periodic security/resilience risk assessments of the operational system performed?

2.8.3: Are periodic penetration testing and vulnerability scanning of the operational system performed to identify vulnerabilities?

2.8.4: Is the behavior of the operational system monitored to identify signs of attack?

2.8.5: Are security/resilience controls monitored during operations and sustainment?

2.8.6: Are confidentiality, integrity, and availability requirements for system data reassessed periodically during operations and sustainment?

2.8.7: Are vulnerabilities, threats, and risks identified and tracked to closure?

2.8.8: Are protection strategies (e.g., program protection plan, security/resilience controls) for the operational system updated periodically or when the threat profile changes?

[45] The *operational system* in this context is the system that has been acquired, developed, and deployed. Once it is operational, a system must be supported and maintained over time.

2.8.9: Is data collected, analyzed, and communicated to provide adequate situational awareness of the operational system's threat environment?

2.8.10: Are changes to the operational system's risk posture reported to the authorizing official in accordance with the monitoring strategy?

2.8.11: Are patches applied to the operational system when appropriate?

2.8.12: Are disruptions that affect the operational system managed?

2.8.13: Are suggested system changes or improvements related to security/resilience communicated to the engineering team?

2.8.14: Is a decommissioning strategy defined for addressing security/resilience concerns when the operational system is removed from service?

2.8.15: Is automation implemented, where feasible, to enable more effective security/resilience risk management during operations and sustainment?

Context

O&S is a lifecycle phase in which a system is used and supported. Its primary focus is managing risk while cost-effectively ensuring that the system successfully supports its missions. Its secondary focus is decommissioning and disposing of the system once it has reached the end of its useful life.

The responsibilities of system operators/maintainers in O&S organizations include managing modifications, upgrades, and future increments of the system. Operators/maintainers also manage changes to system-support artifacts and activities and implement process improvements where appropriate.

When making changes to the system and its system-support artifacts, the following must be considered:

- operational and mission needs
- remaining expected service life
- interoperability issues
- technology improvements
- component obsolescence
- security/resilience issues

Most O&S activities focus on system operation and maintenance; however, an effective risk management capability must also be implemented to do the following:

- identify security/resilience risks
- implement plans to manage the identified risks in accordance with the organization's overall risk tolerance

It is impossible to eliminate all risks; instead, an acceptable level of security/resilience risks can be maintained. Effective risk management during O&S requires establishing a baseline security/resilience configuration for the operational system. Implementing a baseline configuration of

system settings is an initial step toward managing security/resilience risks for the operational system. However, systems operate in dynamic environments, so new vulnerabilities will be discovered in a system's components, new threats will emerge, and patches/updates for software will become available.

Since many patches/updates correct known vulnerabilities in a system and its components, a system's security/resilience risk profile continually changes over time. Managing risk profile changes during O&S requires conducting the following assessment and monitoring practices effectively:

- **Risk assessments** identify security/resilience risks to a system and the missions that it supports. Risk assessments are broad and typically include the analysis of threats, weaknesses, vulnerabilities, access points, and attack paths. The status of each identified risk must be tracked over time, including its priority level and mitigation status. A risk-tracking system reports the mitigation status of identified risks to stakeholders.
- **Vulnerability assessments** identify vulnerabilities in the operational system. Vulnerabilities are typically detected using penetration testing and vulnerability scanning tools. Identified vulnerabilities are tracked, and corrective actions (e.g., applying patches and updates) are implemented as appropriate.
- **System and network monitoring** can provide information about changes to a system's risk profile that occur between risk assessments. Monitoring activities focus on detecting cyber threats and data breaches in systems and networks. Effective monitoring helps organizations detect cyber attacks early and respond to them before they cause damage and disruption.

Security/resilience risks can be addressed only when they are communicated to and understood by decision makers. Communicating with system stakeholders enables them to provide input about prioritizing risks and help develop strategies for managing high-priority risks. The O&S organization must establish a process for communicating risk information to all relevant program/system stakeholders.

Finally, where feasible, automation is implemented to enable more effective security/resilience risk management during O&S. Automation can help reduce issues associated with manual processes, including the potential for errors, slow response times, and communication breakdowns.

Competencies

Knowledge Areas	Skills
- standards, laws, regulations, and policies governing security/resilience during system operations and sustainment - technologies integrated into the operational system - baseline security/resilience configurations for the operational system's devices and software - cybersecurity threats and vulnerabilities - controls for protecting systems, networks, and information - risk assessment methods, tools, and techniques - vulnerability assessment methods, tools, and techniques - threat assessment methods, tools, and techniques - relevant vulnerability data sets - adversarial assessment methods, tools, and techniques	- assessing confidentiality, integrity, and availability (CIA) requirements for operational system data - assessing the operational system for misconfigured devices and nonfunctional protections - performing penetration testing and vulnerability scanning of the operational system - performing security/resilience risk assessments of the operational system - establishing mitigations for identified security/resilience risks - preparing reports that document risk assessment results - managing disruptions that affect the operational system - monitoring the operational system for signs of attack - applying patches to the operational system

Guidance for Practice Questions

2.8.1: Is a baseline security/resilience configuration for the system defined and implemented?

Configuration weaknesses and vulnerabilities are defects or flaws in the way a system and system components are set up. Examples of configuration vulnerabilities include inadequate access controls, unprotected files and directories, unsecured access to administrative accounts, and the use of default settings. These vulnerabilities can be minimized using a baseline configuration (i.e., an agreed-upon description of the security/resilience settings for the system at a given point in time). It helps minimize configuration weaknesses and vulnerabilities in an operational system and serves as the basis for defining and managing changes to security/resilience settings.

A baseline configuration is the initial step toward managing the system's security/resilience risks because it documents the recommended security settings for an operational system, including the following:

- authentication
- authorization
- data protection during storing and transmission
- incident management
- physical security
- monitoring

2.8.2: Are periodic security/resilience risk assessments of the operational system performed?

Systems operate in dynamic environments. New vulnerabilities and threats are discovered over time, affecting the system's security/resilience risk profile. In this dynamic environment, periodic risk assessments help provide situational awareness and facilitate effective risk management.

Risk assessments identify security/resilience risks to a system and the missions it supports. Comprehensive risk assessments typically include analyses of threats, weaknesses, vulnerabilities, access points, and attack paths. The frequency and depth of a risk assessment must be informed by the criticality of the system and its operational environment and mission.

It is helpful to establish criteria (e.g., the addition of system capabilities, changes to connections with other systems, changes to third-party suppliers) that trigger risk assessments or other types of risk reviews. Many organizations are also required to comply with specific security regulations and standards that require periodic risk assessments. Performing regular risk assessments has the following advantages:

- helps ensure continued compliance with regulations and standards
- provides evidence of due diligence in maintaining a secure operational system

2.8.3: Are periodic penetration testing and vulnerability scanning of the operational system performed to identify vulnerabilities?

Penetration testing and vulnerability scanning are important O&S activities. Penetration testing mimics real-world attacks to identify how to circumvent a system's security/resilience controls. It typically involves employing cyber attacks on a system using the same tools and techniques used by attackers [NIST 2008].

Vulnerability scanning is a technique used to identify system assets and associated vulnerabilities [NIST 2008]. Automated tools are used to identify and prioritize known vulnerabilities. These tools typically use a database of known vulnerabilities. Detected vulnerabilities are then logged. Some tools assign risk scores to the detected vulnerabilities.

Mitigation of vulnerabilities (e.g., patching) is considered a routine aspect of vulnerability management, but it poses some risk because patches can impact system performance. As a result, O&S managers must assess and manage the risk introduced by patching a system versus the risk of operating an unpatched system.

The results of penetration testing and vulnerability scanning can be analyzed over time to identify trends; therefore, tracking penetration testing and vulnerability scanning activities and their results is recommended. Vulnerability trend data is also useful input to compliance reporting and security audits.

Penetration testing and vulnerability scanning are often performed annually. However, O&S managers should consider a system's risk and threat environment when determining the frequency and extent of penetration testing and vulnerability scanning. To facilitate vulnerability management

activities, many organizations use automated techniques that they perform more frequently. The results of these automated vulnerability scans can be displayed on a dashboard and used to track a system's vulnerabilities over time.

2.8.4: Is the behavior of the operational system monitored to identify signs of attack?

System monitoring refers to the processes, methods, and tools used for collecting and analyzing system data to detect cyber attacks early and respond to them before they cause damage and disruption. It is a valuable practice for detecting and triaging cyber attacks.

Effective monitoring helps system operators/maintainers identify early signs of intrusion and provides stakeholders with information about changes to a system's risk profile. The extent and sophistication of monitoring activities can vary and should reflect the risk to the system.

System monitoring can also be automated; notifications from these tools can help ensure prompt and consistent levels of response. A system's architecture can include sophisticated monitoring tools and advanced cyber protection techniques, all of which can facilitate a more proactive risk management approach. System monitoring data supports an organization's incident management capability, including its cyber forensic activities.

2.8.5: Are security/resilience controls monitored during operations and sustainment?

Security/resilience controls must be monitored during O&S to determine their effectiveness. Multiple approaches are typically used to monitor security controls, including control assessments, vulnerability assessments, and system monitoring.

A control assessment evaluates technical, physical, and administrative controls to determine the extent to which the security/resilience controls are implemented correctly, operating as intended, and producing the desired outcome [NIST/DOC 2012]. The results of a control assessment are documented in a report that includes information about the effectiveness of the controls and provides recommendations for correcting deficiencies in controls.

A vulnerability assessment is a systematic examination of a system to determine the adequacy of security measures, identify security deficiencies, provide data from which to predict the effectiveness of proposed security measures, and confirm the adequacy of such measures after implementation [NIST/DOC 2020]. Penetration testing and vulnerability scanning are examples of techniques used to assess vulnerabilities. The results of vulnerability assessments provide information about the adequacy of a system's technical controls.

System monitoring is used to detect cyber threats and other signs of unusual or suspicious activity. Monitoring activities are performed during system operation to help maintain security/resilience risks within an acceptable tolerance over time. Analysts use a wide range of monitoring tools (e.g., network security monitoring tools, intrusion detection tools, packet sniffers, managed detection services). Data from system monitoring can provide insight into whether security/resilience controls are achieving their intended objectives.

2.8.6: Are confidentiality, integrity, and availability requirements for system data reassessed periodically during operations and sustainment?

Confidentiality, integrity, and availability (CIA) requirements are the foundation for managing security/resilience, and they merit careful, regular review. The emphasis and rigor applied to CIA varies by the system's criticality and threat environment. System stakeholders are an excellent source of information about CIA requirements.

Periodic reviews of CIA requirements with system stakeholders are invaluable since they invite related, broader discussions about the system's security/resilience requirements. Periodic reassessment of CIA requirements also helps ensure that system data is adequately protected and aligned with evolving security/resilience needs.

2.8.7: Are vulnerabilities, threats, and risks identified and tracked to closure?

Tracking key security/resilience factors (e.g., vulnerabilities, threats, risks) is an essential activity. Ongoing identification of these factors is critical for developing adequate situational awareness that can be used to prioritize the program's security/resilience activities. Tracking these factors is particularly helpful when using analytics to examine data in depth or explore linkages or trends in data.

Trend analysis can be used to identify root causes and inform efforts that extend beyond system patching, including the use of resilient architectures. Reporting on tracking activities ensures that system stakeholders remain informed and accountable for managing vulnerabilities, threats, and risk.

2.8.8: Are protection strategies (e.g., program protection plan, security/resilience controls) for the operational system updated periodically or when the threat profile changes?

A protection strategy defines the approach to managing the security/resilience of an operational system over time. It specifies the activities and system controls that prevent or limit disruptions or risks.

The dynamic nature of threats, vulnerabilities, and risks requires periodic reviews of security/resilience activities and system controls. As a result, protection strategies must also be updated periodically. These updates must be based on reviews of existing security/resilience practices and controls, including how effectively they meet their objectives. Keeping protection strategies current enables an organization to effectively manage changes to a system's risk profiles.

2.8.9: Is data collected, analyzed, and communicated to provide adequate situational awareness of the operational system's threat environment?

Situational awareness refers to an understanding of (1) the cyber threat environment where a system operates, (2) the associated risks and impacts, and (3) the adequacy of the organization's risk mitigation measures. Situational awareness helps the organization collect relevant data about a system's security/resilience risks, integrate that data, and disseminate it to system stakeholders.

Data collection and analysis provides a foundation for an O&S organization's situational awareness capability, producing a common operating picture[46] of the operational system and its environment. A timely common operating picture should be communicated to stakeholders in an appropriate format, enabling operators/maintainers and other personnel to quickly grasp the key elements needed for their effective decision making.

Situational awareness data may include sensitive or classified information; therefore, communication channels for this data must be appropriate for the data's classification level.

2.8.10: Are changes to the operational system's risk posture reported to the authorizing official in accordance with the monitoring strategy?

The ongoing identification, monitoring, and management of risk is a leading practice for managing security/resilience. It is useful to provide regular updates from these ongoing activities to system stakeholders and encourage dialog about changes, priorities, and mitigation decisions. Reporting changes to an operational system's risk posture fosters accountability and transparency within the O&S organization.

These reports also document risk-related activities, decision-making processes, and actions taken to address identified risks. Effective reporting establishes clear lines of responsibility and enables effective communication with stakeholders.

The authorizing official (AO) for a system assumes responsibility for operating that system at an acceptable level of risk and may wish to be included in the dialog about security/resilience risks or simply be notified when material changes occur in the system's risk posture. Overall, reporting changes to the operational system's risk posture to the AO ensures effective risk management, facilitates informed decision making, maintains compliance, and supports a proactive approach to managing security/resilience.

[46] A *common operating picture* is a single display of relevant information that facilitates collaborative planning and enables decision makers to achieve situational awareness.

2.8.11: Are patches applied to the operational system when appropriate?

Patches are software updates that address vulnerabilities within a system or software product. Applying patches to an operational system improves security/resilience, mitigates risks, promotes compliance, enhances stability and performance, maintains supplier support, and ultimately helps protect the organization's assets and reputation. Patch management is critical for keeping security/resilience controls current and protecting an operational system against threats.

A range of variables can influence when it is appropriate to patch a system, including the impact on system performance and operations, applicability of the patch to the software version used, and safety concerns. O&S organizations should establish procedures for determining which software needs to be patched and when, making sure to align the patching process with the system's security/resilience objectives.

2.8.12: Are disruptions that affect the operational system managed?

Security/resilience events and incidents (i.e., disruptions) are occurrences that actually or potentially jeopardize the confidentiality, integrity, or availability attributes of an operational system. System events and incidents must be tracked and managed to resolution; most organizations use a ticketing or similar system to help manage and track events and incidents.

Leading practices for managing disruptions include defining escalation criteria for events and incidents, collecting relevant data, and notifying the right personnel when they occur. These practices help ensure that the organization responds appropriately and in a timely manner. Effective disruption management implements strategies and practices that minimize the impact of disruptions and restore normal system functionality as quickly as possible.

2.8.13: Are suggested system changes or improvements related to security/resilience communicated to the engineering team?

Managing system changes and improvements should be supported by established procedures and ongoing oversight. This management includes tracking system changes and improvements and communicating them to system stakeholders when appropriate. A key challenge is establishing criteria that provide guidance about determining which changes or improvements should be implemented and when they should be implemented.

Communication and coordination are important aspects of managing system changes and improvements during O&S. The O&S organization must establish a process for communicating system changes and improvements to all relevant system stakeholders. Required engineering resources should be included in communications about system changes and improvements, especially if engineering changes or improvements are being requested. Engineering change requests for new system capabilities or updates to existing system capabilities initiate engineering activities (as documented in Goals 2.1 through 2.7).

Failing to communicate system changes or improvements to the engineering team can lead to unaddressed vulnerabilities, inefficient resource allocation, increased operational risks, and delayed responses to emerging threats. Effective communication between the O&S organization and the engineering team is vital to sustaining a secure/resilient operational system.

2.8.14: Is a decommissioning strategy defined for addressing security/resilience concerns when the operational system is removed from service?

Decommissioning a system and transitioning to a new operational capability is an important aspect of the systems lifecycle. A system is decommissioned when it reaches the end of its useful life due to factors such as age, economic feasibility, and relevance of capabilities. Disposal is the removal of a system from its operational environment with the intent of permanently terminating its use [SEBoK 2012]. A system must be disposed of in accordance with legal and regulatory requirements and policies related to safety, security/resilience, and environmental considerations.

As with other activities in the systems lifecycle, system decommissioning requires planning, tracking, and management to ensure the process goes smoothly. The rigor of the decommissioning approach must reflect the risk and nature of the system and the functions it supports. All data must be removed from the system to be decommissioned, and proprietary design components must be removed and destroyed. A well-defined decommissioning plan must address proper data disposal, removal of residual data, adherence to compliance requirements, system access deactivation, and responsible disposal of hardware or equipment.

2.8.15: Is automation implemented, where feasible, to enable more effective security/resilience risk management during operations and sustainment?

Security/resilience risks must be managed as a system is operated, maintained, and used. Automation can be implemented to enable this management to be more effective; it reduces issues associated with manual processes, including the potential for errors, slow response times, and communication breakdowns.

Automation can be incorporated into an organization’s assessment and monitoring practices, facilitating the execution of selected practices. Careful application of automation can streamline operational security/resilience processes and reduce response times.

Domain 3: Engineering Infrastructure

This domain manages security/resilience risks in the engineering, development, test, and training environments. The objectives of Domain 3 are to use software, tools, and technologies that support the program's engineering and development activities and to manage security/resilience risks in the engineering infrastructure. Engineers and developers use a variety of software, tools, and technologies to support their design and development activities. Security/resilience engineering software, tools, and technologies need to be procured, installed, and integrated with the program's existing engineering infrastructure.

The engineering infrastructure is the part of the information technology (IT) infrastructure that supports engineering and development activities performed by personnel from the acquisition program, contractors, and suppliers. As a result, the engineering infrastructure can be an attack vector into the software-reliant system that is being acquired and developed. IT support teams need to ensure that they are applying security/resilience practices when managing the engineering infrastructure to ensure that risk is being managed appropriately. Domain 3 comprises the following two goals:

- **Goal 3.1: Engineering Software, Tools, and Technologies.** Security/resilience engineering software, tools, and technologies are integrated with the engineering infrastructure.
- **Goal 3.2: Infrastructure Operations and Sustainment.** Security/resilience risks in the engineering infrastructure are identified and mitigated.

Goal 3.1: Engineering Software, Tools, and Technologies

Security/resilience engineering software, tools, and technologies are integrated with the engineering infrastructure.

The purpose of this goal is to select and integrate security/resilience engineering software, tools, and technologies with the engineering, development, test, and training environments.

Goal Summary

Engineers and developers use a variety of software, tools, and technologies in the engineering infrastructure to support their design and development activities. Security/resilience engineering software, tools, and technologies are procured, installed, and integrated with the engineering infrastructure according to a plan that is based on formal requirements. However, the operation of these software, tools, and technologies must first be authorized by an authorizing official (AO).

It is important to ensure that those who use the software, tools, and technologies receive appropriate training. Other important activities include ensuring that the software, tools, and technologies are updated as needed; their data is collected and maintained; and they are transitioned (as appropriate) to the operations and sustainment (O&S) organization.

These activities related to software, tools, and technologies form this goal's practice questions, which are summarized in the list below and discussed in detail in the *Guidance for Practice Questions* section on page 115.

List of Practice Questions

3.1.1: Is a plan developed and implemented for incorporating security/resilience engineering software, tools, and technologies in the engineering infrastructure?

3.1.2: Are requirements established for security/resilience engineering software, tools, and technologies across the systems lifecycle?

3.1.3: Are security/resilience engineering software, tools, and technologies selected and procured?

3.1.4: Are security/resilience engineering software, tools, and technologies approved for use by an authorizing official?

3.1.5: Are security/resilience engineering software, tools, and technologies deployed in the engineering infrastructure?

3.1.6: Is training and support (e.g., documentation, help desk, user forums) available for operating, maintaining, and using security/resilience engineering software, tools, and technologies?

3.1.7: Is security/resilience engineering data (e.g., software bill of materials, vulnerabilities, weaknesses, abuse/misuse cases, threats) collected and maintained?

3.1.8: Are changes to security/resilience engineering software, tools, and technologies managed?

3.1.9: Are security/resilience engineering software, tools, and technologies transitioned to the operational support organization as appropriate?

Context

An *engineering infrastructure* includes the software, tools, and technologies that support the design and development of a software-reliant system. Engineering software, tools, and technologies support a wide range of activities across the lifecycle, including requirements specification and management, architecture development and analysis, coding, testing, deployment, and training. The engineering infrastructure also includes a range of technologies that support engineering activities, such as servers, controllers, communications and telecommunications equipment, security components (e.g., intrusion detection systems [IDSs]), and other hardware and peripherals.

The engineering infrastructure is part of the information technology (IT) infrastructure that supports engineering and development activities performed by personnel from the acquisition program, contractors, and suppliers. Management in each of these organizations must ensure that funding and resources are allocated to procuring, implementing, and using these tools and technologies. The IT support team in each organization may need training and support for installing, configuring, and troubleshooting security/resilience engineering software, tools, and technologies; engineers and developers may also need training and support in their use.

The engineering infrastructure includes four key environments:

- The **engineering environment** includes engineering software, tools, and technologies for designing, developing, operating, and maintaining systems. It supports engineering activities across the systems lifecycle, including requirements management, architecture specification and analysis, risk assessment and management, third-party component (TPC) tracking, system deployment, and O&S. Seamless integration of security/resilience software, tools, and technologies into the engineering environment enables security/resilience engineering practices to be executed effectively.
- The **development environment** is a workspace with a collection of software, tools, and technologies used to develop the source code for software components. The environment supports the end-to-end software engineering process, including development, staging, and production. It automates or facilitates processes for creating, testing, debugging, patching, updating, and maintaining software. The suite of security/resilience software, tools, and technologies integrated into the software development environment can include static code analysis, dynamic code analysis, interactive testing, and software composition analysis.
- The **test environment** consists of software, tools, and technologies that enable testers to run test cases that they define. The test environment must mimic the production environment, including hardware configurations, software configurations, operating systems, and databases. From a security/resilience perspective, the test environment allows testers to verify a system's security/resilience requirements and perform vulnerability and adversarial assessments in an operationally relevant environment. Security/resilience testing software, tools, and technologies support the program's vulnerability scanning and penetration testing activities.
- The **training environment** is a platform that individuals use to develop the knowledge, skills, and competencies needed to operate, maintain, and use a system. It can take different forms, including physical classrooms, virtual training labs, and simulated environments. A training environment includes a collection of software, tools, and technologies that are

dedicated to training purposes and are separate from operational (i.e., production) systems and devices. Security/resilience training tools and technologies (e.g., simulations and virtual environments) can be added to the training environment to build the knowledge, skills, and competencies for operating, maintaining, and using the system in a secure/resilient manner.

Competencies

Knowledge Areas	Skills
- lifecycle model being applied - systems engineering processes, methods, and tools - software engineering processes, methods, and tools - risk assessment processes, methods, and tools - procurement processes, methods, and tools - IT concepts and practices - computer components and network architectures - networking concepts and protocols	- specifying requirements for security/resilience engineering software, tools, and technologies - selecting and procuring software, tools, and technologies - installing, configuring, and troubleshooting software applications and tools - installing, configuring, and troubleshooting IT components and devices - collecting and maintaining security/resilience engineering data - managing changes (e.g., patches, updates) to security/resilience engineering software, tools, and technologies

Guidance for Practice Questions

3.1.1: Is a plan developed and implemented for incorporating security/resilience engineering software, tools, and technologies in the engineering infrastructure?

Engineers and developers use a variety of software, tools, and technologies in the engineering infrastructure to support their design and development activities. Selecting, procuring, and implementing these software, tools, and technologies requires careful planning and management. The plan must include one or more objectives, describe the actions required to achieve those objectives, and assign responsibilities for completing the actions.

An effective plan must (1) be flexible and updated as new information is collected or as circumstances change and (2) describe the means for actively engaging stakeholders, gathering their input, and keeping them informed.

3.1.2: Are requirements established for security/resilience engineering software, tools, and technologies across the systems lifecycle?

Requirements for the engineering infrastructure specify the software, tools, and technologies that support security/resilience engineering activities. For example, the engineering team might decide to use model-based systems engineering (MBSE) tools to support architecture development. In that case, the team must specify the MBSE tools that will meet its requirements and identify the supporting technologies (e.g., computers, servers) required for effective implementation. Requirements for security/resilience engineering software, tools, and technologies must be specified for

the engineering, development, test, and training environments (i.e., key environments in the engineering infrastructure).

3.1.3: Are security/resilience engineering software, tools, and technologies selected and procured?

Security/resilience engineering software, tools, and technologies are generally off-the-shelf items that suppliers offer. Procurement is the process of obtaining or purchasing products and services from suppliers. It involves a range of activities, including sourcing, negotiating terms, purchasing items, receiving and inspecting products, and keeping records of all the steps in the process.

Selecting security/resilience engineering software, tools, and technologies must be based on the requirements established for the engineering, development, test, and training environments. In effective procurement, requirements must be compiled and codified collaboratively with stakeholders and included in communications with suppliers.

During procurement, clearly defined requirements must be provided to potential suppliers to ensure they align with project expectations and objectives. Suppliers must be carefully vetted to ensure they (1) can meet the needs of the acquisition program and (2) use appropriate security/resilience development practices that meet or exceed the requirements established by the acquisition program.

3.1.4: Are security/resilience engineering software, tools, and technologies approved for use by an authorizing official?

An AO must actively govern the procurement and deployment of engineering software, tools, and technologies and authorize their operation. An essential criterion for that authorization or sign-off is ensuring that software, tools, and technologies meet their security/resilience requirements.

Information systems operated by or on behalf of the U.S. Department of Defense (DoD) and the U.S. Federal Government are required to receive an authorization to operate (ATO) before they can be deployed. This requirement also applies to software, tools, and technologies in a program's engineering infrastructure.[47]

An ATO is an official management decision made by an AO to allow the operation of software, tools, and technologies and explicitly accept the risk to organizational operations. The ATO process provides organizational accountability by requiring the AO to determine whether security/resilience risks are acceptable based on assessments of controls and risks.

[47] An acquisition program's engineering software, tools, and technologies are deployed in the program's operational computing infrastructure. As a result, engineering software, tools, and technologies must be authorized prior to their deployment.

3.1.5: Are security/resilience engineering software, tools, and technologies deployed in the engineering infrastructure?

Deploying security/resilience engineering software, tools, and technologies involves installing, configuring, and testing them. The goal is to ensure that these resources are available to the engineering team. The engineering, procurement, and IT teams must follow established procedures for managing the selection, procurement, and deployment of software, tools, and technologies.

Automation can streamline the deployment process by minimizing errors and inconsistencies, reducing the time and effort required for deployment, and improving overall efficiency and repeatability. Automation also enables IT personnel to deploy software, tools, and technologies consistently and respond to changes and issues quickly.

3.1.6: Is training and support (e.g., documentation, help desk, user forums) available for operating, maintaining, and using security/resilience engineering software, tools, and technologies?

Proficiency in operating, maintaining, and using security/resilience engineering software, tools, and technologies requires specific knowledge and skills. Various methods (e.g., training, documentation, user forums) can help personnel acquire the needed proficiency. Common training practices include orientations, classroom lectures, role playing, simulations, and mentoring. Documentation, provided in either digital or hard-copy format, includes operator manuals, user guides, white papers, case studies, online help, and quick-reference guides. Other support mechanisms (e.g., a help desk) can provide more specific help to operators/maintainers and users.

Training and support activities are essential to ensure that engineers and developers have the necessary knowledge and skills to use security/resilience engineering software, tools, and technologies. To effectively support security/resilience engineering software, tools, and technologies, operators/maintainers must be proficient in controlling access, managing configurations, and monitoring performance. Training and support activities are essential for preparing operators/maintainers with the necessary skills and knowledge to fulfill their responsibilities. Similarly, engineers and developers must gain proficiency in using security/resilience engineering software, tools, and technologies. For example, engineers must learn how to use MBSE tools that support architecture development. Developers must learn about secure coding standards and how to use code analysis tools.

3.1.7: Is security/resilience engineering data (e.g., software bill of materials, vulnerabilities, weaknesses, abuse/misuse cases, threats) collected and maintained?

Effective use of engineering software, tools, and technologies requires access to accurate and timely security/resilience data (e.g., software bills of materials, vulnerabilities, weaknesses, threats, risks). For example, consider the development of security/resilience requirements for a system. When developing security/resilience requirements, engineers analyze security/resilience risks and establish requirements that appropriately mitigate those risks. Engineers often use

known abuse/misuse cases during the risk identification process. Maintaining a repository of these cases enables engineers to develop robust security/resilience requirements.

Organizing and prioritizing security/resilience engineering data often poses a significant challenge that requires careful management and data-driven analytics. Collaboratively managing the data with internal and external stakeholders is the most effective approach. Since security/resilience engineering data is extensive, organizations must establish processes and procedures for collecting, analyzing, and communicating relevant data to engineers and developers.

3.1.8: Are changes to security/resilience engineering software, tools, and technologies managed?

Changes to engineering software, tools, and technologies are inevitable and expected. Changes include corrections, bug fixes, patches, enhancements, and new releases. Change management involves tracking and handling the changes to security/resilience engineering software, tools, and technologies to minimize disruptions to IT services while the changes are made. The acquisition program, contractors, and suppliers must implement processes and procedures for managing these changes.

3.1.9: Are security/resilience engineering software, tools, and technologies transitioned to the operational support organization as appropriate?

Security/resilience engineering software, tools, and technologies support engineering and development activities for the software-reliant system being acquired and developed. However, engineering activities for the system do not end with its deployment. Engineering changes or improvements can be identified during O&S.

To keep the system operational and secure/resilient, the O&S organization (or its engineering and development support teams) perform engineering and development activities, such as technology insertion, upgrades, and modifications (including code development). As a result, some of the security/resilience engineering software, tools, and technologies used during system design and development are transitioned to the O&S organization. During the transition process, the O&S organization must actively engage with engineers and developers to draw on their expertise, gain their support, and ensure that engineering and development responsibilities for the system during O&S are firmly established.

Goal 3.2: Infrastructure Operations and Sustainment

Security/resilience risks in the engineering infrastructure are identified and mitigated.

The purpose of this goal is to manage security/resilience risks when operating and managing information technology (IT) systems and networks in the engineering infrastructure.

Goal Summary

The engineering infrastructure is the part of the IT infrastructure that supports engineering and development activities performed by personnel from the acquisition program, contractors, and suppliers. IT support teams, as part of operations and sustainment (O&S), apply security/resilience practices when managing the engineering infrastructure.

Threat intelligence and situational awareness data are used to assess and manage security/resilience risks. These risks are managed by first defining and implementing a baseline security/resilience configuration for the engineering infrastructure. The engineering infrastructure is managed by monitoring for unusual activity, managing user access, managing changes to systems and networks, backing up data regularly, and managing disruptions. It is also important to establish and test incident response and service continuity plans and define a decommissioning strategy that incorporates security/resilience concerns.

These infrastructure O&S activities form this goal's practice questions, which are summarized in the list below and discussed in detail in the *Guidance for Practice Questions* section on page 121.

List of Practice Questions

3.2.1: Is a baseline security/resilience configuration for the engineering infrastructure defined and implemented?

3.2.2: Are security/resilience risks in the engineering infrastructure's systems and networks assessed and managed?

3.2.3: Are security/resilience risk management activities for the engineering infrastructure informed by threat intelligence and situational awareness?

3.2.4: Is user access to the engineering infrastructure's data, systems, and networks managed?

3.2.5: Are the engineering infrastructure's systems and networks monitored for unusual activity?

3.2.6: Are changes (e.g., upgrades, updates, patches) to the engineering infrastructure's systems and networks managed?

3.2.7: Is the engineering infrastructure's data backed up periodically?

3.2.8: Are incident response and service continuity plans established and tested for the engineering infrastructure?

3.2.9: Are disruptions that affect the engineering infrastructure managed?

3.2.10: Is a decommissioning strategy defined for addressing security/resilience concerns when the engineering infrastructure is removed from service?

Context

An *IT infrastructure* includes (1) the components needed to operate and manage the organization's IT environment, (2) the services the IT support team provides, and (3) the IT systems and networks[48] that support the organization's mission-related activities. An O&S organization's IT support team performs a range of tasks, including setting up servers, configuring networks, managing databases, and monitoring system performance.

Studies show that the security/resilience of an organization's IT infrastructure is linked to its ability to achieve its mission [DSB 2013]. While organizations are often careful to ensure that mission-critical systems are secure/resilient, they typically do not devote the same level of resources and attention to the IT infrastructure that supports mission-critical systems [DoD 2018].

An *engineering infrastructure* is the part of the IT infrastructure that supports engineering and development activities performed by personnel from the acquisition program, contractors, and suppliers. It supports a wide range of activities across the lifecycle, including requirements specification and management, architecture development and analysis, coding, testing, deployment, and training.

Since the IT infrastructure directly supports an organization's mission, it can be an attack vector into the organization's mission-critical systems and networks [DoD 2022c]. And, from the Security Engineering Framework (SEF) perspective, the engineering infrastructure is also an attack vector into the software-reliant system being acquired and developed. As a result, IT support teams must ensure that they apply security/resilience practices when managing the organization's IT and engineering infrastructures.

Effective management of the engineering infrastructure requires (1) establishing a baseline security/resilience configuration for IT systems and networks and (2) managing changes (typically by the IT support team) that affect the baseline. Change management enables the IT support team to implement planned changes to IT systems and networks (e.g., software, hardware, network settings) effectively, efficiently, and without disruption.

Managing an engineering infrastructure's security/resilience risks requires effective assessment and monitoring practices, such as conducting risk assessments to identify security/resilience risks to IT systems and networks. The status of each identified security/resilience risk must be tracked over time, including its priority level and mitigation status. A risk-tracking system helps personnel manage risks and report them and their mitigation status to stakeholders.

Periodic risk assessments provide snapshots of security/resilience risks that affect the engineering infrastructure. Monitoring IT systems and networks can provide the IT support team with information about changes to security/resilience risks between assessments. These monitoring activities focus on detecting unusual or suspicious activity in IT systems and networks. Effective

[48] IT systems and networks include IT hardware, operating systems, application software, networks, data storage, enterprise software, Internet platforms, and services.

monitoring helps an IT support team detect cyber attacks early and respond to them before they cause damage and disruption.

Competencies

Knowledge Areas

- standards, laws, regulations, and policies governing security/resilience in IT systems and networks
- software, tools, and technologies integrated into the engineering infrastructure
- baseline security/resilience configurations for IT systems and networks
- cybersecurity threats and vulnerabilities
- controls for protecting IT systems and networks
- risk assessment methods, tools, and techniques
- vulnerability assessment methods, tools, and techniques
- threat assessment methods, tools, and techniques
- relevant vulnerability data sets
- service continuity planning and management
- adversarial assessment methods, tools, and techniques

Skills

- assessing confidentiality, integrity, and availability (CIA) requirements for IT systems and networks
- assessing IT systems and networks for misconfigured devices and nonfunctional protections
- performing penetration testing and vulnerability scanning of IT systems and networks
- performing security/resilience risk assessments of IT systems and networks
- establishing mitigations for identified security/resilience risks
- preparing reports that document risk assessment results
- managing disruptions that affect IT systems and networks
- monitoring IT systems and networks for signs of attack
- applying patches to IT systems and networks

Guidance for Practice Questions

3.2.1: Is a baseline security/resilience configuration for the engineering infrastructure defined and implemented?

A baseline configuration is an agreed-upon description of the security/resilience settings for IT systems and networks. It documents the recommended security settings for IT systems and networks, including authentication, authorization, data protection during storage and transmission, incident management, physical security, and monitoring.

Configuration vulnerabilities are defects or flaws in the way a system and system components are set up. Examples of such vulnerabilities are inadequate access controls, unprotected files and directories, unsecured access to administrative accounts, and the use of default settings. A baseline configuration helps minimize configuration vulnerabilities in IT systems and networks and serves as the basis for defining and managing changes to security/resilience settings. It is the initial step that IT support teams take to manage security/resilience risks in the engineering infrastructure.

3.2.2: Are security/resilience risks in the engineering infrastructure's systems and networks assessed and managed?

The engineering infrastructure operates in a dynamic environment. New vulnerabilities will be discovered in IT systems and networks, new threats will emerge, new technologies will be implemented, and patches and updates will become available. In this ever-changing environment, it is important to perform periodic security/resilience risk assessments of the engineering infrastructure to gain situational awareness and enable effective IT support and management.

Security/resilience risk assessments typically include analyses of threats, weaknesses, vulnerabilities, access points, and attack paths. It is helpful to establish criteria that trigger risk assessments or other types of risk reviews (e.g., changes in IT policies or the IT infrastructure, newly discovered vulnerabilities and threats).

Many organizations must comply with security regulations and standards that require periodic risk assessments. Regular risk assessments help ensure continued compliance with these regulations and standards and provide evidence of due diligence in maintaining a secure/resilient engineering infrastructure.

3.2.3: Are security/resilience risk management activities for the engineering infrastructure informed by threat intelligence and situational awareness?

Threat intelligence is a collection of threat information that has been aggregated, transformed, analyzed, interpreted, or enriched to provide decision makers with context about potential risks [NIST 2016]. Threat intelligence reports provide important input to an organization's situational awareness activities.

Situational awareness—an understanding of the (1) cyber threat environment where systems and networks operate, (2) associated risks and impacts, and (3) adequacy of its risk mitigation measures—makes it easier to collect relevant data about security/resilience threats and risks to IT systems and networks, integrate that data, and disseminate it stakeholders. Data collection and analysis provides a foundation for the IT support team's situational awareness, producing a common operating picture[49] of the engineering infrastructure and its environment. A timely common operating picture must be communicated to stakeholders in an appropriate format to help the IT support team and other stakeholders quickly understand the key elements needed to assess risks and make effective decisions.

[49] A *common operating picture* is a single display of relevant information that facilitates collaborative planning and enables decision makers to achieve situational awareness.

3.2.4: Is user access to the engineering infrastructure's data, systems, and networks managed?

Access control grants or denies specific requests for (1) obtaining and using information and related information-processing services and (2) accessing physical facilities [NIST/DOC 2020]. The goal of access control is to regulate who or what can view or use resources. It is a fundamental concept in security/resilience management that minimizes risk to an organization and its systems and networks. There are two types of access control:

- **Physical** access control limits access to buildings, rooms, and physical IT assets.
- **Logical** access control limits network access to data, systems, and networks in the engineering infrastructure.

Two key elements of logical access control are authentication and authorization:

- **Authentication** verifies the identity of a user, process, or device, often as a prerequisite to allowing access to resources in a system [NIST/DOC 2020].
- **Authorization** adds an extra layer of security to the authentication process. It is used to grant access privileges to a user, program, or process [NIST/DOC 2020]. Authorization determines whether a user should be granted access to data or make a specific transaction.

Access to the engineering infrastructure must be managed to ensure that engineers and developers access only the data, systems, and networks needed to perform their assigned tasks.

3.2.5: Are the engineering infrastructure's systems and networks monitored for unusual activity?

System and network monitoring refers to the processes, methods, and tools used to collect and analyze data to detect cyber attacks early and respond to them before they cause damage and disruption. Effective monitoring helps the IT support team identify early signs of intrusion and provides stakeholders with information about changes to systems and networks in the engineering infrastructure. The extent and sophistication of monitoring activities can vary and should reflect the risk to the software-reliant system being acquired, designed, and developed.

An IT organization's monitoring practice can also be automated to enable notifications from automated monitoring tools and help ensure prompt and consistent levels of response. System and network monitoring generates data that supports an organization's incident management capability, including its cyber forensic activities.

3.2.6: Are changes (e.g., upgrades, updates, patches) to the engineering infrastructure's systems and networks managed?

IT change management involves planning and implementing changes to IT systems and networks. An IT support team must address several types of routine changes to the engineering infrastructure, including implementing the following planned changes:

- An **upgrade** implements major changes to the engineering infrastructure, such as increasing the functionality of software, adding new hardware, or modifying the network architecture. Upgrades can increase the lifespan and efficiency of IT systems and networks.
- An **update** is new, improved, or fixed software that replaces older versions of the same software.
- A **patch** is a specific type of update that addresses vulnerabilities in a system or software product. Patch management is critical for maintaining security/resilience controls and protecting systems and software against threats.

A variety of variables can influence how and when changes to the engineering infrastructure are implemented, including the following:

- the impact on system and network performance
- the ability to reduce security/resilience risks
- safety concerns

Implementing planned changes improves security/resilience, mitigates risks, promotes compliance, enhances stability and performance, maintains supplier support, and ultimately helps protect the organization's assets and reputation.

3.2.7: Is the engineering infrastructure's data backed up periodically?

Data backup involves making a copy of digitized data in case the data is damaged, deleted, or lost. Backup copies are stored in a separate physical or virtual location in case of equipment failure or another catastrophe. The backup copy is used to recover or restore data when needed.

Data backup is an important activity that decreases the risk of full or partial data loss when unexpected events occur by enabling the organization to restore systems and applications to a previous state. Data backup is a key component of an organization's service continuity plan. To help ensure that backups are functioning as intended, they must be regularly tested for their viability.

3.2.8: Are incident response and service continuity plans established and tested for the engineering infrastructure?

Security/resilience events and incidents (i.e., disruptions) affect the confidentiality, integrity, or availability attributes of IT systems and the data that they process, store, and transmit. Managing disruptions that affect the engineering infrastructure includes two core activities:

- **Incident response** involves monitoring and detecting events and incidents in an IT infrastructure and executing proper responses to them. An incident response plan outlines the predefined instructions/procedures for detecting, responding to, and limiting the consequences of cyber attacks on an organization's information systems [NIST 2010].
- **Service continuity management** involves contingency planning for recovery when IT systems or networks are damaged or rendered unavailable. It supports an organization's business continuity management plan by ensuring that IT systems and networks are able to provide acceptable levels of services during and after major disruptions. A service continuity plan provides an organization with predefined procedures for sustaining IT services during adverse conditions, ranging from minor interruptions to large-scale incidents.

Program personnel, contractors, and suppliers must collaborate to develop incident response and service continuity plans to protect the engineering infrastructure and sustain IT services. Involving key stakeholders when regularly testing these plans is essential to maintain their completeness and effectiveness. Testing also helps ensure that the IT support team is ready to execute its responsibilities as outlined in these plans.

3.2.9: Are disruptions that affect the engineering infrastructure managed?

A *disruption* is an event or incident that affects the confidentiality, integrity, or availability attributes of IT systems and the data that they process, store, and transmit. Incident response and service continuity plans establish approaches for managing cyber disruptions. The acquisition program, contractors, and suppliers must manage cyber events and incidents according to these plans.

Events and incidents must be tracked and managed to resolution. Many organizations use a ticketing or similar system to streamline this process. Leading practices for managing disruptions include defining escalation criteria for events and incidents, collecting relevant data, and notifying the right personnel when events and incidents occur. These practices help the organization respond appropriately and in a timely manner. Effective disruption management involves implementing strategies and practices that minimize the impact of disruptions and restore normal system functionality as quickly as possible.

3.2.10: Is a decommissioning strategy defined for addressing security/resilience concerns when the engineering infrastructure is removed from service?

The engineering infrastructure supports engineering and development activities performed by an acquisition program, contractors, and suppliers. It includes engineering software, tools, and technologies that support the acquisition, design, and development of a system. However, engineering

software, tools, and technologies are not necessarily delivered to the O&S organization when a system is deployed [NIST 2021]. As a result, the acquisition program, contractors, and suppliers must decommission the software, tools, and technologies that are not being transitioned to the O&S organization.

Disposal of engineering software, tools, and technologies must adhere to applicable laws, regulations, and policies related to safety, security/resilience, and environmental considerations. Decommissioning demands planning, tracking, and management to ensure the process goes smoothly. The rigor of the decommissioning approach must reflect the risk and nature of the system and the functions it supports. All data must be removed from software, tools, and technologies, and proprietary design components must be removed and destroyed. A well-defined decommissioning plan must address proper data disposal, removal of residual data, adherence to compliance requirements, system access deactivation, and responsible disposal of hardware or equipment.

Part 3
Appendices

This part of the report provides background information about SEF acronyms, glossary terms, and references. This supplementary material is helpful in understanding SEF concepts presented throughout the report.

Appendix A: Acronyms

AI
Artificial Intelligence

AO
Authorizing Official

ASF
Acquisition Security Framework

ATO
Authorization to Operate

BOM
Bill of Materials

cATO
Continuous Authorization to Operate

CDR
Critical Design Review

CIA
Confidentiality, Integrity, and Availability

CNSS
Committee on National Security Systems

COTS
Commercial-Off-the-Shelf

CVE
Common Vulnerabilities and Exposures

DAST
Dynamic Application Security Testing

DATO
Denial of Authorization to Operate

DevSecOps
Development, Security, and Operations

DoD
Department of Defense

DT&E
Developmental Test and Evaluation

GOTS
Government-Off-the-Shelf

IATT
Interim Authorization to Test

IDS
Intrusion Detection System

IT
Information Technology

ITRA
Independent Technical Risk Assessment

MBSE
Model-Based Systems Engineering

MDAP
Major Defense Acquisition Program

ML
Machine Learning

MLOC
Million Lines of Code

MOSA
Modular Open System Approach

NDA
Non-Disclosure Agreement

NIST
National Institute of Standards and Technology

NIST RMF
National Institute of Standards and Technology Special Publication 800-37, *Risk Management Framework for Information Systems and Organizations*

O&S
Operations and Sustainment

OSS
Open Source Software

OT&E
Operational Test and Evaluation

PDR
Preliminary Design Review

POA&M
Plan of Action and Milestones

PPP
Program Protection Plan

PROM
Programmable Read-Only Memory

RAM
Reliability, Availability, Maintainability

RAR
Risk Assessment Report

RMF
Risk Management Framework

ROM
Read-Only Memory

RTM
Requirements Traceability Matrix

SAF
Software Assurance Framework

SAR
Security Assessment Report

SAST
Static Application Security Testing

SBOM
Software Bill of Materials

SCRM
Supply Chain Risk Management

SDLC
Systems development lifecycle or software development lifecycle

SEF
Security Engineering Framework

SEI
Software Engineering Institute

SERA
Security Engineering Risk Analysis

SME
Subject Matter Expert

SoS
System of Systems

SRR
System Requirements Review

T&E
Test and Evaluation

TEMP
Test and Evaluation Master Plan

TPC
Third-Party Component

TRR
Test Readiness Review

Appendix B: Glossary

Accept
An option for managing or handling a risk where, if a risk occurs, its consequences will be tolerated and no proactive action to address the risk will be taken (When a risk is accepted, the rationale for doing so is documented.)

Access Control
The process of granting or denying specific requests for (1) obtaining and using information and related information processing services and (2) accessing physical facilities [NIST/DOC 2020] (The objective is to regulate who or what can view or use resources in a computing environment.)

Acquisition Enterprise
The broader organization in which the acquisition program resides (It typically provides services that support program execution, such as operational test and evaluation, authorization to operate, operations and sustainment, and independent assessments. Some acquisition enterprises also provide security/resilience subject matter experts to assist program personnel.)

Acquisition Program
An organizational unit responsible for acquiring and developing a software-reliant system in response to a defined need (Program personnel manage engineering activities across the lifecycle and perform selected system and software engineering activities. An acquisition program is also referred to as a *program*.)

Adapt
A strategy for controlling security/resilience risks that provides a sustained capability to accommodate changes in a system's risk environment, including changes to threats, vulnerabilities, mission, and technologies

Administrative Control
Policies, procedures, or guidelines that define personnel and organizational practices in accordance with the organization's security goals (Examples of administrative controls include security education training and awareness programs, password management policies, and incident response planning.)

Adversarial Assessment
An assessment that evaluates the system's security/resilience using realistic tactics, techniques, and procedures while in a relevant operating environment (These assessments are planned and coordinated attacks by trained experts on a system or technology environment.)

Agile
An iterative approach to system design and development where lengthy requirements, build, and test phases are allocated to smaller work increments, allowing software increments to be delivered more frequently

Analytic Approach
An approach that defines how to assess security/resilience risks that includes (1) an assessment approach (i.e., quantitative, qualitative, semi-quantitative) and (2) an analysis approach (i.e., threat oriented, asset/impact oriented, vulnerability oriented) [NIST/DOC 2012].

Analyze
A risk management activity where probability, impact, and risk exposure are evaluated for each risk and those measures are used to establish risk handling priorities (The objective is to gain a better understanding of risks by examining risk-related data in relation to a set of evaluation criteria.)

Architect (SEF Role)
An individual who defines the architecture for a software-reliant system in order to meet specified requirements

Architecture
The structure of components, their relationships, and the principles and guidelines governing their design and evolution over time [DAU 2024] (See *system architecture* or *software architecture*.)

Architecture Risk Analysis
An analysis that identifies and corrects design weaknesses and the risks they pose to the mission (This analysis is a foundational aspect of a program's security/resilience engineering activities.)

Architecture Tradeoff Analysis
An analysis that enables architects and engineers to identify and prioritize design decisions across multiple quality attributes, including security/resilience (Tradeoffs are evaluated among candidate design options. Each option's impact on the system's quality attributes is evaluated, and the option that best addresses overall system requirements is selected.)

Assess
The process of identifying and analyzing risks

Assessment Team (SEF Role)
The collection of assessors who work together to evaluate a system

Assessor (SEF Role)
An individual who conducts an assessment of a program or system to evaluate performance

Attack
See *Cyber Attack*.

Attack Path
A pathway or method that an attacker uses to access a system and exploit the system's weaknesses and vulnerabilities

Attack Surface
The set of points on the boundary of a system, a system component, or an environment where an attacker can try to enter, cause an effect on, or extract data from that system, system component, or environment [NIST/DOC 2020]

Attack Vector
See *Attack Path*.

Attack-Path Analysis
An analysis that examines the potential pathways that adversaries can use to access a system (The results of an attack-path analysis can be used to reduce a system's attack surface by limiting or removing unnecessary access points or services that are identified as weaknesses or vulnerabilities.)

Authentication
The process of verifying the identity of a user, process, or device, often as a prerequisite to allowing access to resources in a system [NIST/DOC 2020]

Authorization
The process of granting access privileges to a user, program, or process [NIST/DOC 2020] (The authorization process determines whether a user should be granted access to data or make a specific transaction.)

Authorization Boundary
All the components of a system that will be included in assessment and authorization activities, establishing the scope for the authorization-to-operate process (The authorization boundary excludes separately authorized systems with which the system is connected as well as enabling systems that provide support across the systems lifecycle [e.g., development, test, and training environment].)

Authorization Decision
A formal statement by the authorizing official regarding the acceptance of the risk associated with operating a system and expressed as an authorization to operate, interim authorization to test, or denial of authorization to operate

Authorization Package
Documentation that includes the control assessment report, risk assessment report, and Plan of Action and Milestones (The package provides the authorizing official with the information required to make a risk-based decision about whether to authorize the operation of the system.)

Authorization to Operate (ATO)
1. A formal declaration by an authorizing official that a system is approved to operate at an acceptable level of risk based on the implementation of an approved set of technical, managerial, and procedural safeguards
2. The official management decision given by a senior organizational official to authorize operation of an information system and to explicitly accept the risk to organizational operations (including mission, functions, image, or reputation), organizational assets, personnel, other organizations, and the nation based on the implementation of an agreed-upon set of security/resilience controls [NIST/DOC 2011]

Authorizing Official (AO)
An individual with the authority to formally assume responsibility for operating a system at an acceptable level of risk to agency operations, agency assets, or personnel [NIST 2022] (The term *designated official* is used in National Institute of Standards and Technology documentation when referring to an authorizing official.)

Availability
An attribute that ensures timely and reliable access to and use of information [NIST/DOC 2020]

Avoid
An option for managing or handling a risk where activities are restructured to eliminate the possibility of a risk occurring

Baseline Configuration
1. An agreed-upon description of the security/resilience controls for the system at a given point in time
2. The recommended security/resilience settings for information technology systems and networks (Examples of security/resilience settings include authentication, authorization, data protection during storage and transmission, incident management, physical security, and monitoring.)

Baseline Control Selection
A selection of processes that use predefined sets of controls, called control baselines, as the starting point for selecting security/resilience controls

Bill of Materials (BOM)
A list of the hardware, software, firmware, and service components included in the system (These components can be monitored over time for issues, vulnerabilities, and updates.)

Black-Box Testing
A method of testing that examines the functionality of software without knowing its internal code structure, implementation details, and internal paths (*Black-box testing* is also known as *functional testing*, *behavioral testing*, and *closed-box testing*.)

Blue Team Assessment
A type of assessment that is performed proactively to prevent problems from occurring (These assessments are typically scheduled well in advance of a program milestone [e.g., reviews, deliveries] to provide a program with sufficient time to implement recommended changes and prepare for that milestone successfully. Security/resilience blue teams focus on maintaining security/resilience defenses against all cyber attacks, threats, and risks [NIST/DOC 2020].)

Change Management
A practice for managing and tracking modifications and updates to software, tools, technologies, and processes, including implementing upgrades, updates, and patches

Code Review
A disciplined engineering practice used for detecting and correcting defects in software code (During a code review, programmers systematically check each other's code for errors, irregular formatting, or inconsistencies with system requirements that may lead to problems during software integration. Code reviews generally examine four key areas: defects in the code, consistency with the system's software requirements and design, quality of documentation, and consistency with coding standards.)

Common Control
A type of security/resilience control that can be inherited by one or more systems across an enterprise (A set of common controls is identified across the enterprise where the acquisition program resides. Common controls are allocated to organizational entities designated as *common control providers*.)

Common Operating Picture
A single display of relevant information that facilitates collaborative planning and enables decision makers to achieve situational awareness

Communicate
An interactive process of exchanging risks and concerns among individuals and groups across the program [DoD 2017]

Compliance Assessment
A type of assessment that is used to establish how well a program or organization is complying with a standard, policy, or regulation (From a security/resilience perspective, a compliance assessment is often used to provide an independent review of security/resilience controls selected for the system.)

Component
One of multiple elements that make up a system or subsystem (Common types of components include hardware, software, firmware, and services.)

Condition–Consequence Statement
A format used to document risk statements (The *condition* part of the statement establishes the circumstances that are causing concern, while the *consequence* part expresses the resulting losses if the risk is realized [Dorofee 1996].)

Confidentiality
An attribute that preserves authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information [NIST/DOC 2020]

Configuration Vulnerabilities
Defects or flaws in the way information technology systems and networks are set up (Inadequate access controls, unprotected files and directories, unsecured access to administrative accounts, and use of default settings are examples of configuration vulnerabilities.)

Consequence
The loss that results when a threat exploits one or more vulnerabilities (The loss is measured in relation to the status quo [i.e., current state].)

Continuous Authorization to Operate (cATO)
Authorization of the platform, process, and team that produces the system under a continuous monitoring process and maintains the residual risk within the risk tolerance of the authorizing official

Contractor
An organization that enters into an agreement, (i.e., contract) with an acquisition program to provide custom components to the program (Contractors are responsible for many engineering and development activities, including requirements specification, architecture development, and implementation and coding activities.)

Control
A safeguard or countermeasure prescribed for an information system or an organization to protect the confidentiality, integrity, and availability of the system and its information [NIST/DOC 2020]

Control Assessment
An assessment that evaluates technical, physical, and administrative controls to determine the extent to which the security/resilience controls are implemented correctly, operating as intended, and producing the desired outcome [NIST/DOC 2012] (The results of a control assessment are documented in a report that includes information about the effectiveness of the controls and provides recommendations for correcting deficiencies in controls.)

Control Baseline
A predefined set of controls that can be used as the starting point for selecting security/resilience controls

Critical Design Review (CDR)
An event to ensure that the design meets the requirements

Custom Component
A component that has been specifically designed or adapted for use in a system or subsystem and is not generally available in the marketplace

Cyber Attack
Actions taken using computer networks to disrupt, disable, destroy, or maliciously control a computing environment/infrastructure; destroy the integrity of the data; or steal controlled information [NIST/DOC 2012]

Cyber-Physical System
An engineered system that is built from and depends on the seamless integration of computational algorithms and physical components (Cyber-physical systems merge the physical and virtual worlds, integrating objects, data, and services [NIST 2022].)

Data Backup
The process of making a copy of digitized data in case the data is damaged, deleted, or lost (Copies of digitized data are stored in a separate physical or virtual location for preservation in case of equipment failure or catastrophe.)

Data Repository
A structure used to collect, manage, and store data (It often refers to a specific setup, such as a group of databases or a data warehouse.)

Denial of ATO (DATO)
An indication that there are major weaknesses or deficiencies in the security/resilience controls employed within or inherited by a system

Deployment
A lifecycle phase in which a system and its components are installed, tested, and implemented in an operational environment (System deployment typically involves the transition of system capabilities to end users and the transition of support and maintenance responsibilities to the post-deployment support organization [SEBoK 2023].)

Deployment Plan
A plan that documents the activities needed to manage the transition of the system to its intended operational environment, including key roles and responsibilities for internal and external stakeholders (including suppliers)

Design Weakness
A defect or flaw in a system's architecture or detailed design with the potential for exploitation when implemented

Detect
A strategy for controlling security/resilience risks where the occurrence of a security/resilience threat (i.e., cyber attack) is identified via program or system monitoring activities

Developer (SEF Role)
The individual who designs and writes software code, builds software components, or tests software performance

Development Environment
A workspace with a collection of software, tools, and technologies used to develop the source code for software components (The development environment supports the end-to-end software engineering process, including development, staging, and production. It automates or facilitates processes for creating, testing, debugging, patching, updating, and maintaining software.)

Developmental Test & Evaluation (DT&E)
Verification that the system is built correctly in accordance with the specification and contract

DevOps
An iterative engineering approach where development (Dev) and operations (Ops) teams work collaboratively across the lifecycle to increase the speed and quality of deployed software

DevSecOps
An iterative engineering approach where development (Dev), operations (Ops), and security (Sec) teams work collaboratively across the lifecycle to increase the speed, quality, and security of deployed software

Direct Consequence
The immediate result of a cyber attack (Direct consequences produce one or more of the following outcomes regarding confidentiality, integrity, and availability requirements: unauthorized information disclosure or information theft [confidentiality], modification or manipulation of information or services [integrity], destruction of information [availability], and interruption of access to information or services [availability].)

Disruption

1. A security/resilience event or incident that actually or potentially jeopardizes the confidentiality, integrity, or availability attributes of an operational system
2. An event or incident that affects the confidentiality, integrity, or availability attributes of information technology systems and the data that they process, store, and transmit

Dynamic Application Security Testing (DAST)

A form of black-box testing using dynamic code analysis to test the binary code of running software for potentially exploitable vulnerabilities (Dynamic analysis tools identify both compile-time and runtime vulnerabilities, such as configuration errors that appear only within a realistic execution environment.)

Dynamic Code Analysis

See *Dynamic Application Security Testing (DAST)*.

Emergent Behavior

A characteristic (i.e., behavior) of a system of systems that arises from the interactions among the individual systems and is not embodied in any of the individual systems

End User (SEF Role)

An individual who ultimately uses or is intended to use the system
See *User*.

End-User Organization

The organization that includes the individuals who ultimately use or are intended to use the system (The end-user organization stands in contrast to the operations and sustainment organization that supports and maintains the system.)

Engineer (SEF Role)

The individual who develops and manages user, system, and software requirements; leads the development of the system architecture; evaluates design tradeoffs; or oversees verification and validation activities

Engineering Environment

The environment that includes engineering software, tools, and technologies for designing, developing, operating, and maintaining systems (The engineering environment supports engineering activities across the systems lifecycle.)

Engineering Infrastructure
The software, tools, and technologies in the information technology infrastructure that support engineering and development activities performed by personnel from the acquisition program, contractors, and suppliers (The engineering infrastructure supports a wide range of activities across the lifecycle, including requirements specification and management, architecture development and analysis, coding, testing, deployment, and training.)

Engineering Lead for the Program (SEF Role)
The individual who provides technical expertise and strategic leadership for the design and development of complex software-reliant systems, including specifying requirements, defining architectures, and performing verification and validation activities

Engineering Project Plan
A plan that establishes how the engineering team will perform security/resilience engineering activities across the lifecycle (The engineering project plan identifies the tasks, personnel [for both program and contractor personnel], budget, and schedule for implementing the activities in the selected lifecycle model.)

Engineering Risk Management Plan
A plan that defines the activities that a program intends to perform when assessing and managing security/resilience risks for the system of interest

Engineering Team (SEF Role)
The collection of engineers developing a system

Enterprise
An organization with a defined mission and a defined boundary that uses information systems to execute its mission and is responsible for managing its own risks and performance

Evaluate
An activity performed during risk analysis to establish probability, impact, and risk exposure measures for each identified risk (The evaluate activity is performed during risk analysis.)

Evaluation Criteria
A set of measures for assessing the probability and impact during qualitative risk analysis

Evolutionary Character
A characteristic of a system of systems that grows and changes independently of other systems over time

Exploit
Software or a sequence of commands that takes advantage of a vulnerability to cause unintended or unanticipated behavior to occur in a system or its components

Firmware
Computer programs and data stored in hardware's read-only memory or programmable read-only memory that provides instructions on how the hardware/device is supposed to operate

Functional Architecture
An architecture that defines the system's functions and how they interact with each other to achieve the system's mission (The functional architecture provides the foundation for the system architecture through the allocation of functions and subfunctions to system components, facilities, and processes [DoD 2022e].)

General-Purpose System
A system that is designed to be user friendly, run a variety of applications, and support multiple users and devices (Examples of general-purpose systems include mainframe computers, servers, laptops, desktop computers, smartphones, and tablets.)

Geographic Distribution
A characteristic of a system of systems that reflects the dispersal of individual systems within a system of systems over large geographic areas

Handle/Treat
A risk management activity where a plan to address or handle each risk is developed and implemented based on available mitigation options (accept, avoid, transfer, watch, mitigate) (The objective of handling a risk is to take proactive action to address high-priority risks before they become problems.)

Hardware
The material physical components of a system

Identify
A risk management activity where concerns and uncertainties are transformed into distinct, tangible risks that are documented in a prescribed format (The objective is to anticipate what could go wrong.)

If-Then Statement
A specific format used to document a risk statement (The *if* part of the statement conveys how the threat exploits a vulnerability, while the *then* portion expresses the resulting consequence.)

Impact
A measure of the loss that occurs when a risk is realized (Impact is one of the three risk measures along with probability and risk exposure.)

Incident Response
An activity that involves the monitoring and detection of events and incidents in an information technology infrastructure and the execution of proper responses to those events and incidents

Incident Response Plan
A plan that documents a predetermined set of instructions or procedures to detect, respond to, and limit consequences of a malicious cyber attack against an organization's information system(s) [NIST 2010]

Increment
A software release that provides capabilities, features, and/or planned changes (For most systems, software is deployed in multiple releases or increments.)

Independent Assessment
An assessment that is performed by individuals who are not connected with a program or system (An independent assessment provides an objective, unbiased review of a program or system. Examples of independent assessments include programmatic assessments, technical assessments, compliance assessments, process assessments, blue team assessments, and red team assessments.)

Independent Technical Risk Assessment (ITRA)
A type of assessment that the Department of Defense is required to perform on major defense acquisition programs prior to milestone and production decisions (Independent technical risk assessments provide a view of program technical risk that is independent of the program and its chain of command.)

Indirect Consequences
The impacts on other systems and mission threads/workflows (Indirect consequences determine the extent to which a cyber attack will lead to mission degradation or mission failure. Assessors use indirect consequences to establish a risk's impact.)

Information Technology Infrastructure
The components needed for the operation and management of an organization's information technology environment and the services it provides (The information technology infrastructure includes the information technology systems and networks that support an organization's mission-related activities.)

Information Technology Support Group (SEF Role)
The group that manages the information technology systems and networks (i.e., the engineering infrastructure) that support engineering and development activities, including setting up servers, configuring networks, managing databases, and monitoring system performance

Integrity
An attribute that guards against improper information modification or destruction and includes ensuring information non-repudiation and authenticity [NIST/DOC 2020]

Interim Authorization to Test (IATT)
Temporary authorization to test an information system in a specified operational information environment within the time frame and under the conditions or constraints enumerated in the written authorization [CNSS 2022]

Lifecycle Model
A conceptual representation used in project management to describe how development activities are implemented in a given development project

Logical Access Control
A control that limits network access to data, systems, and networks in the engineering infrastructure (Two key elements of logical access control are authentication and authorization.)

Manage (Risk Management)
The process of handling and tracking risks over time

Managerial Independence
A characteristic of a system of systems where the management of each system within a system of systems is independent from the management of the other systems

Mission
1. A set of objectives and goals to be achieved in a specific operational environment [DoD 2020b]
2. A specific task that an individual or a group is assigned to perform (An individual or group must perform several activities or steps in pursuit of a mission.)

Mission Thread
A sequence of end-to-end activities and events presented as a series of steps to achieve an objective or goal (e.g., a mission) [DoD 2020b] (Synonyms include *workflow*, *work process*, and *business process*.)

Mitigate
An option for managing or handling a risk where action is taken to reduce or contain a risk (Mitigation can include implementing controls/countermeasures to address a risk.)

Mitigation Plan
A strategy or set of actions for addressing a risk based on available mitigation options (e.g., accept, avoid, transfer, watch, mitigate) (Mitigation plans for controlling security/resilience risks are based on the following strategies: protect [reduce threats and minimize consequences], detect [identify signs of threat occurrence], respond [take action on detected threats], recover [restore access to and functionality], and adapt [accommodate changes in a system's risk environment].)

Non-Disclosure Agreement (NDA)
A signed form that is used to protect sensitive and confidential information from being disclosed by recipients of that information (An individual might be asked to sign a non-disclosure agreement and agree that they will not disclose sensitive information to others.)

Off-the-Shelf Component
A system component that is available as a stock item or commodity (An off-the-shelf component is not specially designed or custom made. Off-the-shelf software components include commercial-off-the-shelf software, government-off-the-shelf software, and open source software.)

Operational Independence
A characteristic of a system of systems where each system within a system of systems provides useful functionality apart from other systems

Operational Risk
The risk of loss resulting from inadequate or failed internal processes, policies, systems, or events that disrupt business operations

Operational Test and Evaluation (OT&E)
A test that validates that the system can successfully accomplish its mission in a relevant operational environment (OT&E is performed by an organization that is independent of the acquisition program.)

Operationally Relevant Environment
The set of operational conditions, selected by system users in coordination with the testing organization, that represent the range of a system's operations (Operationally relevant environments include a broad range of options, including models, simulations, testbeds, prototypes, full-scale engineering development models of the system, and the actual environment where the system is deployed.)

Operations and Sustainment (O&S)
A lifecycle phase in which a system is used and supported

Operations and Sustainment (O&S) Manager (SEF Role)
The individual who oversees the operation and maintenance of a deployed system

Operations and Sustainment (O&S) Organization

The organization responsible for operating and maintaining deployed systems (The operations and sustainment organization manages modifications, upgrades, and future increments of the system. It also manages changes to system support artifacts and activities and implements process improvements where appropriate.)

Operator/Maintainer

An individual serving in an information technology role who monitors and controls computer systems, ensures that machines and computers are running correctly, and performs routine and preventative maintenance (e.g., general computer tasks, backups/restores, software loads, disk utilization, system loading, system monitoring)

Operator/Maintainer (SEF Role)

An individual who monitors, supports, troubleshoots, and maintains system components to ensure that the system is functioning and meeting the needs of end users

Organization-Generated Control

A strategy where an organization uses its own process to select security/resilience controls

Patch

A specific type of software update that addresses vulnerabilities within a system or software product (Patch management is critical for keeping security/resilience controls current and protecting systems and software against threats.)

Peer Review

See *Code Review*.

Penetration Testing

A test methodology in which assessors, typically working under specific constraints, attempt to circumvent or defeat the security features of a system [NIST/DOC 2020]

Physical Access Control

A type of physical control that limits access to buildings, rooms, and physical information technology assets

Physical Architecture

An architecture that "consists of one or more product structures, or views, of the physical solution" [DoD 2022e] (A physical architecture illustrates how the system's physical elements and interfaces are arranged. It documents the design for the system consistent with the functional architecture and system requirements.)

Physical Control
A collection of mechanisms that deny unauthorized access to facilities, equipment, and resources, and protects personnel and property from damage or harm (Examples of physical controls are card readers, cameras, motion sensors, intruder alarms, equipment inventories, surge protectors, and fire protection.)

Plan of Action and Milestones (POA&M)
A document that identifies tasks that need to be accomplished (A Plan of Action and Milestones details resources required to accomplish the elements of the plan, milestones for meeting the tasks, and the scheduled completion dates for the milestones [NIST/DOC 2020].)

Preliminary Design Review (PDR)
An event to ensure that the planned technical approach (i.e., system architecture) will meet the requirements

Prime Contractor
The entity responsible for managing system development, including organizing and managing subcontractors

Prioritize
The process of determining the order in which risks should be addressed, based on probability, impact, and risk exposure measures

Probability
A measure of the likelihood that the risk will occur (Probability is one of the three risk measures along with impact and risk exposure.)

Process Assessment
A review typically performed as part of an organizational improvement initiative (Process assessments typically assess a program or organization in relation to a standard set of criteria or a model. The main purpose of a process assessment is to facilitate process improvement activities for a program or organization.)

Procurement
The process of obtaining or purchasing products and services from suppliers (Procurement involves a range of activities, including sourcing, negotiating terms, purchasing items, receiving and inspecting products, and keeping records of all the steps in the process.)

Product Risk Management
The process of managing security/resilience risks in the system that is being designed and developed, focusing on risks resulting from the exploitation of vulnerabilities in a system and its components (*Product risk management* is a synonym of *system risk management*.)

Program
See *Acquisition Program*.

Program Manager (SEF Role)
An individual with responsibility for and authority to accomplish program objectives for the development, production, and sustainment of the system to meet the user's operational needs

Program Protection Plan (PPP)
A plan that is used to coordinate and integrate all security efforts throughout the systems lifecycle, ensuring that a program's technology, components, and information are protected adequately [DoD 2011]

Program/System Stakeholder (SEF Role)
An individual, group, or organization with a vested interest (i.e., stake) in the decision making and activities of the acquisition program or about the system that is being acquired, developed, and operated

Programmatic Assessment
A type of assessment that provides a comprehensive and systematic review of an acquisition program's managerial and technical progress (A programmatic assessment is typically designed to identify program cost, schedule, and performance risks; formulate risk mitigation plans; and provide feedback to program stakeholders.)

Project Risk Management
The process of identifying and managing project-level security/resilience risks, including risks related to resources, cost, and schedule

Protect
A strategy for controlling security/resilience risks so that vulnerability to threats is reduced and any consequences or losses that might occur are minimized

Protection Strategy
An approach used to manage the security/resilience of an operational system over time (The protection strategy specifies the organizational activities and system controls that prevent or limit disruptions or risks.)

Provenance
The chronology of the origin, development, ownership, location, and changes to a system or system component and its associated data (Provenance may also include the personnel and processes used to interact with or make modifications to the system, component, or associated data [NIST 2020].)

Qualitative Risk Evaluation

An approach based on the assignment of descriptors such as low, medium, and high to evaluate risk [NIST 2020] (Evaluation criteria establish a set of qualitative measures for assessing the probability and impact of risk. A risk exposure matrix provides a way of estimating the magnitude of a risk based on the current values of probability and impact.)

Quality Attributes

A set of functional and nonfunctional requirements that are used to evaluate system performance [Wang 2023] (Examples of quality attributes include security/resilience, reliability, interoperability, usability, portability, maintainability, and scalability.)

Quantitative Risk Evaluation

An approach that assigns numerical values to impact and probability based on statistical probabilities and the valuation of loss or gain [NIST 2020] (Quantitative risk analysis provides more objective data than qualitative analysis because it is based on measurable risk data.)

Real-Time System

A system in which computation must be performed during the actual time that an external process occurs, allowing computational results to respond to those external processes [DAU 2024] (Examples of real-time systems include industrial control systems, automobile-engine fuel injection systems, medical imaging systems, command-and-control systems, and weapon systems.)

Recover

A strategy for controlling security/resilience risks where access to and functionality of a system (or systems) are restored after a risk's consequences, losses, and damages are realized

Red Team Assessment

A type of assessment that is performed when a program is experiencing significant, unanticipated problems or when program management suspects that risks or issues need immediate attention to keep the program on track (Security/resilience red teams are often chartered as adversarial exercises designed to compromise organizational missions or business processes [NIST/DOC 2020].)

Reference Design

A technical blueprint of a system that is intended for others to copy (A reference design defines the essential elements of the system. Organizations using a reference design can enhance or modify the design as needed to meet organization-specific requirements. The *DoD Enterprise DevSecOps Reference Design* [DoD 2019] leverages a set of hardened DevSecOps tools and deployment templates that enable DevSecOps teams to select the appropriate template for the program application capability to be developed [DoD 2010].)

Repository

See *Data Repository*.

Requirement
A documented necessary attribute, capability, characteristic, or quality of a system that benefits stakeholders

Requirements Inspection
A review of requirements to find defects, such as missing requirements, ambiguous requirements, inconsistent requirements, or flawed assumptions (Requirements inspections can be done at varying levels of formality, from error/defect inspections to peer reviews.)

Requirements Traceability Matrix (RTM)
A matrix that provides a logical way of capturing requirements, establishing where they came from and what standards apply, and documenting changes as the development activities progress through the systems lifecycle

Resilience
The ability to prepare for and adapt to changing conditions and recover rapidly from disruption (Resilience includes the ability to withstand and recover from deliberate attacks, accidents, or naturally occurring threats or incidents [NIST/DOC 2020].)

Resources
The assets needed to produce a deliverable, helping a program progress from initial concept to the delivery of products and services (Three critical resources for security/resilience engineering are funding, staffing, and tools.)

Respond
A strategy for controlling security/resilience risks where actions are taken to counteract a detected threat (i.e., cyber attack) and minimize consequences, losses, and damages

Risk (General Definition)
The probability of suffering harm or loss [Dorofee 1996]

Risk (Security/Resilience Definition)
1. A measure of (1) the likelihood that a threat will exploit a vulnerability to produce an adverse consequence or loss and (2) the magnitude of the loss [Alberts 2014]
2. "A measure of the extent to which an entity is threatened by a potential circumstance or event, and typically is a function of: (i) the adverse impact, or magnitude of harm, that would arise if the circumstance or event occurs; and (ii) the likelihood of occurrence" [CNSS 2022]

Risk Analysis
See *Analyze*.

Risk Assessment
A systematic process of identifying and analyzing the risks to an activity or undertaking (From a systems perspective, a risk assessment identifies, evaluates, and prioritizes security/resilience risks for a system that is being acquired and developed.)

Risk Assessment Context
The operational environment in which a system will be deployed, including the mission threads supported by the system and interfaces with other independently managed systems

Risk Assessment Report (RAR)
1. A report that documents the results of the risk assessment performed during the authorization-to-operate process (The risk assessment report documents risk data generated during the assessment, including threats, vulnerabilities, predisposing conditions, likelihood of occurrence for threats, magnitude of impact for threats, risks, and risk management strategies.)
2. A report that contains the results of performing a risk assessment or the formal output from the process of assessing risk [CNSS 2022]

Risk Assessment Scope
The boundary conditions of the risk assessment, establishing what is and is not included in the assessment (An assessment's scope is affected by several factors, including stakeholder needs, resources, and time available.)

Risk Communication
An interactive process of exchanging risks and concerns among individuals and groups across the program (The purpose of risk communication is to help personnel understand the system's security/resilience risks and the strategies being implemented to manage them.)

Risk Escalation
A formal process that transfers ownership and accountability for a risk or issue to a higher authority or stakeholder (Escalating project risks and issues requires informing relevant program stakeholders about the potential or actual problems that need their attention or intervention.)

Risk Evaluation
See *Evaluate*.

Risk Exposure
A measure of the magnitude of a risk based on current values of probability and impact (Risk exposure is one of the three risk measures along with probability and impact.)

Risk Identification
See *Identify*.

Risk Management

1. A systematic approach for minimizing exposure to potential losses (Risk management provides a disciplined environment for continuously assessing what could go wrong [i.e., assessing risks], determining which risks to address [i.e., setting mitigation priorities], and implementing actions to address high-priority risks and bring those risks within tolerance.)
2. The program and supporting processes to manage security/resilience risks to organizational operations, organizational assets, individuals, other organizations (Activities for managing security/resilience risks include (1) establishing the context for risk-related activities, (2) assessing risks, (3) managing risks, and (4) monitoring risks over time [NIST/DOC 2012].)

Risk Mitigation

See *Mitigate*.

Risk Model

A model that establishes the risk factors to be assessed and the relationships among the factors (Common risk factors included in risk assessments are threat, vulnerability, impact, likelihood, and predisposing conditions [NIST/DOC 2012].)

Risk Posture

A system's defense against cyber attacks, including the overall management and strategy of protecting the system, its components, and the data that it stores, processes, and transmits

Risk Prioritization

See *Prioritize*.

Risk Statement

A statement that provides a succinct and unique description of each security/resilience risk (The description uses a simple notation to document each risk and its consequences if the risk is realized. This approach can facilitate risk tracking.)

Secure Coding Standards

Language-specific rules and guidelines (e.g., for C, C++, Java, Perl) used to prevent security weaknesses (Secure coding standards include guidance for adopting naming conventions, eliminating buffer overflows, and bounds checking variables.)

Security

A condition that results from the establishment and maintenance of protective measures that enable an organization to perform its mission or critical functions despite risks posed by threats to its use of systems (Protective measures may involve a combination of deterrence, avoidance, prevention, detection, recovery, and correction that should form part of the organization's risk management approach [NIST/DOC 2020].)

Security Assessment Report (SAR)

1. A report that documents the results of the control assessment performed during the authorization-to-operate process (A security assessment report (1) documents any deficiencies in controls that were not resolved during the development of the system or that were discovered after system development and (2) provides a list of recommended corrective actions for any weaknesses or deficiencies identified in the security controls.)
2. A report that provides a disciplined and structured approach for documenting the findings of the assessor and the recommendations for correcting any identified vulnerabilities in the security controls [CNSS 2022]

Security/Resilience

A term used by the SEF to refer to both security and resilience as part of its discussion of risk (See the definitions of both *Security* and *Resilience*.)

Security/Resilience Events and Incidents

See *Disruption*.

Security/Resilience Requirement

A specified capability or need that a system must satisfy to mitigate security/resilience risks

Security/Resilience Subject Matter Expert (SME) (SEF Role)

An individual who has knowledge and expertise in security/resilience technical and management activities

Service

Infrastructure, platform, or software that is hosted or provided by a third party

Service Continuity Management

A process that involves contingency planning for recovery when information technology systems or networks are damaged or made unavailable (Service continuity management supports an organization's business continuity management by ensuring that information technology systems and networks are able to provide acceptable levels of information technology services during and after major disruptions.)

Service Continuity Plan

A plan that documents predefined procedures for sustaining information technology services during adverse conditions, ranging from minor interruptions to large-scale incidents

Situational Awareness

An understanding of (1) the cyber threat environment where a system operates, (2) the associated risks and impacts, and (3) the adequacy of its risk mitigation measures

Software
A set of instructions, data, or programs used to operate computers and execute specific tasks

Software Architecture
The set of structures needed to reason about the system, which comprises software elements, relations among them, and properties of both [Clements 2010] (A software architecture documents the high-level design of a software component. It represents the design decisions related to the overall structure and behavior of a system's software components.)

Software Assurance
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 [CNSS 2010]

Software Bill of Materials (SBOM)
A list of the software components that are integrated into the system (A software bill of materials facilitates the management of software components, including tracking licenses and vulnerabilities. It also documents an inventory of a system's software dependencies.)

Software Code
A series of instructions that perform a particular task (The logic in the software code enables a computer to perform specified functions.)

Software Integration
The combination of multiple software units into one system, subsystem, or component (The level of integration can range from combining relatively small pieces of code into an integrated component to combining several large software components into a system of significant size and complexity.)

Software Provenance
The collection of verifiable information regarding the origin of software, including its history of ownership, development, and distribution (From a security/resilience perspective, software provenance provides information for tracking software defects, vulnerabilities, and dependencies.)

Software-Reliant System
A system whose behavior (e.g., functionality, performance, safety, security, interoperability) is dependent on software in some significant way [Bergey 2009] (Any system where software influences the system's design, construction, deployment, and evolution is considered to be a software-reliant system.)

Static Application Security Testing (SAST)

A form of white-box testing (Static code analysis tools attempt to identify weaknesses in static [i.e., non-running] source code. Static analysis identifies issues within the code's logic and techniques. This approach is often used to detect security weaknesses, performance issues, non-compliance with standards, and outdated programming constructs.)

Static Code Analysis

See *Static Application Security Testing (SAST)*.

Subsystem

An integrated set of components that interact with the greater system to perform a function (A subsystem must be integrated with other subsystems and components to make a system. A complex system can be decomposed into subsystems with specified subsystem interfaces. The designation of subsystems is often based on information flows within the system.)

Supplier

An organization that provides off-the-shelf components that are integrated into the system that is being acquired and developed (Off-the-shelf components delivered by suppliers include commercial-off-the-shelf software, government-off-the-shelf software, and open source software.)

System

1. An arrangement of components that work together to achieve a given purpose or provide a needed capability [SEF definition]
2. "An aggregation of system elements and enabling system elements to achieve a given purpose or provide a needed capability" [DoD 2022e]
3. "A combination of interacting elements organized to achieve one or more stated purposes" [NIST/DOC 2020]
4. An entity that interacts with other entities (i.e., other systems), including hardware, software, humans, and the physical world (The function of a system establishes what the system is intended to do and is described by the functional specification in terms of functionality and performance [Avizienis 2004].)

System Architecture

An architecture that documents the structure of system components, their relationships, and the principles and guidelines governing their design and evolution over time (It is common for a system architecture to include a functional architecture and a physical architecture.)

System Component

See *Component*.

System Control
The collection of security/resilience safeguards or countermeasures that are implemented at the system level (A control assessment determines whether system controls are implemented correctly, operating as intended, and producing the desired outcome.)

System Decommissioning
The retirement or termination of a system and its operations (A system is decommissioned when it reaches the end of its useful life due to factors such as age, economic feasibility, and relevance of capabilities.)

System Deployment
See *Deployment*.

System Disposal
The removal of a system from its operational environment with the intent of permanently terminating its use [SEBoK 2012] (A system must be disposed of in accordance with legal and regulatory requirements and policies related to safety, security/resilience, and environmental considerations.)

System Integrator
An entity that assembles components and subsystems and ensures that the system functions as a whole

System Monitoring
Collecting and analyzing system data using processes, methods, and tools to detect cyber attacks early and respond to them before they cause damage and disruption (The extent and sophistication of system monitoring activities can vary and should reflect the risk faced by the system.)

System of Interest
The focus of the systems engineering effort [NIST 2021] (*System of interest* refers to the system that is being acquired and developed and is undergoing an assessment. It includes system components, system component interconnections, and the environment where they are placed [NIST 2021].)

System of Systems (SoS)
A set or arrangement of interdependent systems that are related or connected (i.e., networked) to provide a given capability [Levine 2003]

Systems Lifecycle
In the *Security Engineering Framework*, the activities involved in acquiring and developing a software-reliant system from initial concept through system disposal

Technical Assessment
A type of assessment that focuses on solving technical problems, answering technical questions, or supporting technical decisions (A technical assessment provides an objective, technical evaluation of a system and its underlying technologies. The assessment team members must be experts in the technology and domain of the system.)

Technical Control
The collection of safeguards or countermeasures that are primarily implemented and executed through mechanisms contained in system components [NIST/DOC 2012] (Firewalls, intrusion detection systems, encryption, and identification and authentication mechanisms are examples of technical controls.)

Technical Interchange Meeting
A meeting between relevant program/system stakeholders to discuss technical information about a system's design and development (Technical interchange meetings are scheduled periodically across the systems lifecycle.)

Technical Reviews
Events that enable engineering and program personnel to evaluate significant achievements and assess technical maturity and risk [DAU 2023] (The program manager and engineering team use technical reviews to define and control the program's technical effort.)

Test and Evaluation (T&E)
A lifecycle phase where systems or system components are examined in relation to their requirements and specifications

Test and Evaluation Master Plan (TEMP)
A plan that documents an acquisition program's planned testing activities (The Test and Evaluation Master Plan is the overarching document for managing a program's test and evaluation activities.)

Test Environment
The software, tools, and technologies that enable testers to run test cases (The test environment must mimic the production environment, including hardware configurations, software configurations, operating systems, and databases.)

Tester (SEF Role)
An individual who plans, prepares, and executes tests of a system to (1) validate whether the system meets its requirements and (2) verifies that the system fulfills its intended purpose

Third-Party Assessor
An assessment team from an organization without connections to or relationships with the acquisition program, its parent enterprise, and its contractors

Third-Party Component (TPC) Providers
Contractors, suppliers, and other external organizations that develop custom and off-the-shelf components

Third-Party Components (TPCs)
Off-the-shelf components (i.e., hardware, software, firmware, services) provided by contractors, suppliers, and other external organizations

Threat
A cyber-based act, occurrence, or event that exploits one or more vulnerabilities and leads to an adverse consequence or loss

Threat Modeling
"A form of risk assessment that models aspects of the attack and defense sides of a logical entity, such as a piece of data, an application, a host, a system, or an environment" [NIST/DOC 2020] (Threat modeling is used to identify use cases, threat agents, attack vectors and patterns, design patterns, and compensating controls needed to mitigate risk.)

Track
A risk management activity where risks and mitigation plans are monitored over time (The objectives are to identify and manage (1) the effectiveness of mitigation plans, (2) changes that can affect a risk's measures, and (3) conditions indicating that new risks may have emerged.)

Training Environment
A platform, including software, tools, and technologies, that individuals use to develop the knowledge, skills, and competencies needed to operate, maintain, and use a system (The training environment can take different forms, including physical classrooms, virtual training labs, and simulated environments.)

Training Initiative
An initiative that provides educational activities designed to improve the job performance of an individual or group (Training initiatives typically involve advancing an individual's knowledge and skills to enhance job performance. Common training practices include orientations, classroom lectures, case studies, role playing, simulations and computer-based training, and mentoring.)

Transfer
An option for managing or handling a risk where a risk is shifted to another party (e.g., through insurance or outsourcing)

Treat
See *Handle/Treat*.

User
An individual who utilizes computing resources when performing their job duties

User (SEF Role)
See *End User*.

Vulnerability
1. A defect in a system that has known exploits
2. A defect in a software, firmware, hardware, or service component resulting from a weakness that can be exploited to produce a negative impact to the confidentiality, integrity, or availability of system components and their associated data [MITRE 2023]
3. "A known weakness in a system, system security procedures, internal controls, or implementation by which an actor or event may intentionally exploit or accidentally trigger the weakness to access, modify, or disrupt normal operations of a system—resulting in a security incident or a violation of the system's security policy" [CNSS 2022]
4. "A weakness in an information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source" [NIST/DOC 2020]

Vulnerability Assessment
A systematic examination of an information system or product to determine the adequacy of security measures, identify security deficiencies, provide data from which to predict the effectiveness of proposed security measures, and confirm the adequacy of such measures after implementation [NIST/DOC 2020]

Vulnerability Scanning
See *Vulnerability Assessment*.

Watch
An option for managing or handling a risk where a risk is monitored for changes to its risk measures (impact, probability, risk exposure)

Waterfall
A sequential model for system development (The Waterfall model is divided into phases, where the output of one phase becomes the input of the next phase. A given phase in the lifecycle must be complete before the next phase starts. The phases in a Waterfall lifecycle do not overlap.)

Weakness

1. A defect or flaw in a system, maintenance procedure, internal control, architecture, design, or implementation with the potential to be exploited by a threat actor
2. A condition in a software, firmware, hardware, or service component that, under certain circumstances, could contribute to the introduction of vulnerabilities [MITRE 2023] (Examples of software weaknesses include buffer overflows, structure and validity problems, channel and path errors, authentication errors, resource management errors, insufficient verification of data, and code evaluation and injection.)
3. An attribute or characteristic that may, under known or unknown conditions, render an entity, asset, system, network, or geographic area open to exploitation or susceptible to a given hazard [CNSS 2022]
4. A defect or characteristic that may lead to undesirable behavior [NIST 2022]

White-Box Testing

A method of testing that examines the software's internal structure, design, and coding (The purpose of white-box testing is to verify input/output flows and improve design, usability, and security. *White-box testing* is also known as *clear-box testing*, *glass-box testing*, *transparent-box testing*, and *structural testing*.)

Workflow

A collection of interrelated work tasks that achieves a specific result [Sharp 2008] (A workflow includes all tasks, procedures, organizations, personnel, technologies, tools, data, inputs, and outputs required to achieve the desired objectives. Synonyms include *mission thread*, *work process*, and *business process*.)

Appendix C: References

[Alberts 2002]
Alberts, Christopher & Dorofee, Audrey. *Managing Information Security Risks: The OCTAVE[SM] Approach.* Addison-Wesley. 2002. ISBN 978-0-321-11886-8. https://insights.sei.cmu.edu/library/managing-information-security-risks-the-octave-approach/

[Alberts 2014]
Alberts, Christopher; Woody, Carol; & Dorofee, Audrey. *Introduction to the Security Engineering Risk Analysis (SERA) Framework.* CMU/SEI-2014-TN-025. Software Engineering Institute, Carnegie Mellon University. 2014. https://insights.sei.cmu.edu/library/introduction-to-the-security-engineering-risk-analysis-sera-framework/

[Alberts 2017a]
Alberts, Christopher & Woody, Carol. *Prototype Software Assurance Framework (SAF): Introduction and Overview.* CMU/SEI-2017-TN-001. Software Engineering Institute, Carnegie Mellon University. 2017. https://insights.sei.cmu.edu/library/prototype-software-assurance-framework-saf-introduction-and-overview/

[Alberts 2017b]
Alberts, Christopher; Woody, Carol; Wallen, Charles; & Haller, John. Assessing DoD System Acquisition Supply Chain Risk Management. *CrossTalk*. Volume 30. Issue 3. May/June 2017. Page 4. https://insights.sei.cmu.edu/library/assessing-dod-system-acquisition-supply-chain-risk-management/

[Alberts 2022]
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. https://insights.sei.cmu.edu/library/acquisition-security-framework-asf-managing-systems-cybersecurity-risk/

[Avizienis 2004]
Avizienis, Algirdas; Laprie, Jean-Claude; Randell, Brian; & Landwehr, Carl. Basic Concepts and Taxonomy of Dependable and Secure Computing. *IEEE Transactions on Dependable and Secure Computing.* Volume 1. Issue 1. January–March 2004. Pages 11–33. https://ieeexplore.ieee.org/document/1335465

[Bergey 2009]
Bergey, John K. *A Proactive Means for Incorporating a Software Architecture Evaluation in a DoD System Acquisition.* CMU/SEI-2009-TN-004. Software Engineering Institute, Carnegie Mellon University. 2009. https://insights.sei.cmu.edu/library/a-proactive-means-for-incorporating-a-software-architecture-evaluation-in-a-dod-system-acquisition/

[Charette 1990]
Charette, Robert N. *Application Strategies for Risk Analysis.* McGraw-Hill Book Company. 1990. ISBN 978-0-07-010888-2. https://dl.acm.org/doi/10.5555/102899

[Clements 2010]
Clements, Paul; Bachmann, Felix; Bass, Len; Garlan, David; Ivers, James; Little, Reed; Merson, Paulo; Nord, Robert; & Stafford, Judith. *Documenting Software Architectures: Views and Beyond.* Addison-Wesley. 2010. ISBN 978-0-321-55268-6. https://insights.sei.cmu.edu/library/documenting-software-architectures-views-and-beyond-second-edition/

[CNSS 2010]
Committee on National Security Systems (CNSS). *National Information Assurance (IA) Glossary.* CNSSI No. 4009. CNSS. 2010. https://www.dni.gov/files/NCSC/documents/nittf/CNSSI-4009_National_Information_Assurance.pdf

[CNSS 2022]
Committee on National Security Systems (CNSS). *Committee on National Security Systems (CNSS) Glossary.* CNSSI 4009. CNSS. March 2, 2022.[50]

[DAU 2023]
Defense Acquisition University (DAU). *Systems Engineering Brainbook: Technical Reviews and Audits*. 2023. https://content1.dau.edu/DAUMIG_se-brainbook_189/content/Technical_Reviews_and_Audits.html

[DAU 2024]
Defense Acquisition University (DAU). *DAU Glossary of Defense Acquisition Acronyms and Terms*. March 28, 2024 [accessed]. https://www.dau.edu/glossary

[DoD 2010]
Department of Defense (DoD). *Reference Architecture Description.* DoD. June 2010. https://dodcio.defense.gov/Portals/0/Documents/Ref_Archi_Description_Final_v1_18Jun10.pdf

[DoD 2011]
Department of Defense (DoD). *Program Protection Plan Outline & Guidance, Version 1.0.* DoD. 2011. https://acqnotes.com/wp-content/uploads/2018/04/PPP-Outline-and-Guidance-v1-July2011.pdf

[50] To access the glossary, visit the CNSS Instructions page (https://www.cnss.gov/CNSS/issuances/Instructions.cfm), scroll to CNSSI 4009, and follow the link.

[DoD 2017]
Department of Defense (DoD). *Department of Defense Risk, Issue, and Opportunity Management Guide for Defense Acquisition Programs.* DoD. 2017. https://acqnotes.com/wp-content/uploads/2017/07/DoD-Risk-Issue-and-Opportunity-Management-Guide-Jan-2017.pdf

[DoD 2018]
Department of Defense (DoD). *Weapon Systems Cybersecurity: DOD Just Beginning to Grapple with Scale of Vulnerabilities.* GAO-19-128. U.S. Government Accountability Office (GAO). 2018. https://www.gao.gov/products/gao-19-128

[DoD 2019]
Department of Defense (DoD). *DoD Enterprise DevSecOps Reference Design.* DoD. 2019. https://dodcio.defense.gov/Portals/0/Documents/DoD%20Enterprise%20DevSecOps%20Reference%20Design%20v1.0_Public%20Release.pdf

[DoD 2020a]
Department of Defense (DoD). *Defense Technical Risk Assessment Methodology (DTRAM): Criteria Volume, Version 6.3.* DoD. 2020. https://ac.cto.mil/wp-content/uploads/2021/01/DTRAM-0-1.pdf

[DoD 2020b]
Department of Defense (DoD). *Mission Engineering Guide.* Department of Defense (DoD). 2020. https://ac.cto.mil/wp-content/uploads/2020/12/MEG-v40_20201130_shm.pdf

[DoD 2022a]
Department of Defense (DoD). *Risk Management Framework for DoD Systems*. DoD Instruction 8510.01. DoD. July 19, 2022. https://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodi/851001p.pdf

[DoD 2022b]
Department of Defense (DoD). *Cyber Survivability Endorsement (CSE) Implementation Guide, Version 2.0*. DoD. August 22, 2022. https://www.dau.edu/cop/rqmt/documents/guide-cyber-survivability-endorsement-implementation-guide

[DoD 2022c]
Department of Defense (DoD). *DoD Cybersecurity: Enhanced Attention Needed to Ensure Cyber Incidents Are Appropriately Reported and Shared.* GAO-23-105084. U.S. Government Accountability Office (GAO). 2022. https://www.gao.gov/products/gao-23-105084

[DoD 2022d]
Department of Defense (DoD) *OSD Memorandum for Senior Pentagon Leadership Defense Agency and DoD Field Activity Directors.* DoD. February 4, 2022. https://dodcio.defense.gov/Portals/0/Documents/Library/20220204-cATO-memo.PDF

[DoD 2022e]
Department of Defense (DoD). *Systems Engineering Guidebook.* DoD. February 2022. https://ac.cto.mil/wp-content/uploads/2022/02/Systems-Eng-Guidebook_Feb2022-Cleared-slp.pdf

[Dorofee 1996]
Dorofee, Audrey; Walker, Julie; Alberts, Christopher; Higuera, Ron; Murphy, Richard; & Williams, Ray. *Continuous Risk Management Guidebook.* Software Engineering Institute, Carnegie Mellon University. 1996. https://insights.sei.cmu.edu/library/continuous-risk-management-guidebook/

[DSB 2013]
Defense Science Board (DSB). *Task Force Report: Resilient Military Systems and the Advanced Cyber Threat.* Office of the Under Secretary of Defense for Acquisition, Technology, and Logistics OUSD(AT&L). 2013. https://apps.dtic.mil/sti/citations/ADA569975

[Dvorak 2009]
Dvorak, Daniel. NASA Study on Flight Software Complexity. *AIAA Infotech at Aerospace Conference and Exhibit.* April 6, 2009. https://doi.org/10.2514/6.2009-1882

[Hilburn 2023]
Hilburn, Tom; Fairley, Dick; & Squires, Alice. Software Engineering in the Systems Engineering Life Cycle. *SEBoK Website.* December 14, 2023 [accessed]. https://sebokwiki.org/wiki/Software_Engineering_in_the_Systems_Engineering_Life_Cycle

[Kloman 1990]
Kloman, H. Felix. Risk Management Agonists. *Risk Analysis.* Volume 10. Number 2. June 1990. Pages 201–205. https://onlinelibrary.wiley.com/doi/10.1111/j.1539-6924.1990.tb01039.x

[Levine 2003]
Levine, Linda; Morris, Edwin J.; Place, Patrick R. H.; Plakosh, Daniel; & Meyers, B. Craig. *Proceedings of the System of Systems Interoperability Workshop (February 2003).* CMU/SEI-2003-TN-016. Software Engineering Institute, Carnegie Mellon University. June 2003. https://insights.sei.cmu.edu/library/proceedings-of-the-system-of-systems-interoperability-workshop-february-2003/

[Maier 1996]
Maier, Mark. Architecting Principles for Systems-of-Systems. Pages 567–574. *Proceedings of the Sixth Annual International Symposium of INCOSE.* July 1996. https://doi.org/10.1002/j.2334-5837.1996.tb02054.x

[MITRE 2023]
The MITRE Corporation. Common Weakness Enumeration (CWE). *MITRE Website.* December 14, 2023 [accessed]. https://cwe.mitre.org/

[NIST 2008]
National Institute of Standards and Technology (NIST). *Technical Guide to Information Security Testing and Assessment.* NIST SP 800-115. NIST. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-115.pdf

[NIST 2010]
National Institute of Standards and Technology (NIST). *Contingency Planning Guide for Federal Information Systems.* NIST SP 800-34r1. NIST. 2010. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-34r1.pdf

[NIST 2016]
National Institute of Standards and Technology (NIST). *Guide to Cyber Threat Information Sharing.* NIST SP 800-150. NIST. 2016. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-150.pdf

[NIST 2018]
National Institute of Standards and Technology (NIST). *Framework for Improving Critical Infrastructure Cybersecurity, Version 1.1.* NIST. 2018. https://doi.org/10.6028/NIST.CSWP.04162018

[NIST 2020]
National Institute of Standards and Technology (NIST). *Integrating Cybersecurity and Enterprise Risk Management (ERM).* NISTIR 8286. NIST. 2020. https://doi.org/10.6028/NIST.IR.8286

[NIST 2021]
National Institute of Standards and Technology (NIST). *Developing Cyber-Resilient Systems: A Systems Security Engineering Approach.* NIST SP 800-160v2r1. NIST. December 2021. https://csrc.nist.gov/publications/detail/sp/800-160/vol-2-rev-1/final

[NIST 2022]
National Institute of Standards and Technology (NIST). *Engineering Trustworthy Secure Systems.* NIST SP 800-160v1r1. NIST. November 2022. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-160v1r1.pdf

[NIST/DOC 2011]
Joint Task Force Transformation Initiative: National Institute of Standards and Technology (NIST) and U.S. Department of Commerce (DOC). *Managing Information Security Risk: Organization, Mission, and Information System View.* NIST SP 800-39. NIST. March 2011. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-39.pdf

[NIST/DOC 2012]
Joint Task Force Transformation Initiative: National Institute of Standards and Technology (NIST) and U.S. Department of Commerce (DOC). *Guide for Conducting Risk Assessments.* NIST SP 800-30r1. NIST. September 2012. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-30r1.pdf

[NIST/DOC 2018]
Joint Task Force Interagency Working Group: National Institute of Standards and Technology (NIST) and U.S. Department of Commerce (DOC). *Risk Management Framework for Information Systems and Organizations: A System Life Cycle Approach for Security and Privacy.* NIST SP 800-37r2. NIST. December 2018. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-37r2.pdf

[NIST/DOC 2020]
Joint Task Force Interagency Working Group: National Institute of Standards and Technology (NIST) and U.S. Department of Commerce (DOC). *Security and Privacy Controls for Information Systems and Organizations.* NIST SP 800-53r5. NIST. September 2020. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf

[Novak 2023]
Novak, William E. What's Going On in My Program? 12 Rules for Conducting Assessments [blog post]. *SEI Blog*. June 19, 2023. https://insights.sei.cmu.edu/blog/whats-going-on-in-my-program-12-rules-for-conducting-assessments/

[Pratt 2023]
Pratt, Mary K. Project Management. *TechTarget Website.* December 14, 2023 [accessed]. https://www.techtarget.com/searchcio/definition/project-management

[SEBoK 2012]
Systems Engineering Body of Knowledge (SEBoK) Governing Board. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, Version 3.2.2.* INCOSE-TP-2003-03.2.2. International Council on Systems Engineering (INCOSE). 2012. https://sebokwiki.org/wiki/INCOSE_Systems_Engineering_Handbook

[SEBoK 2023]
Systems Engineering Body of Knowledge (SEBoK) Governing Board. Guide to the Systems Engineering Body of Knowledge (SEBoK). *SEBoK Website*. November 20, 2023. https://sebokwiki.org/wiki/Guide_to_the_Systems_Engineering_Body_of_Knowledge_(SEBoK)

[Sharp 2008]
Sharp, Alec & McDermott, Patrick. *Workflow Modeling: Tools for Process Improvement and Application Development.* Artech House. 2008. https://us.artechhouse.com/Workflow-Modeling-Tools-for-Process-Improvement-and-Application-Development-Second-Edition-P1215.aspx

[U.S.C. 2013]
One Hundred Twelfth Congress of the United States of America (U.S.C.). *National Defense Authorization Act for Fiscal Year 2013.* H.R. 4310. United States Government Publishing Office (GPO). 2013. http://www.gpo.gov/fdsys/pkg/BILLS-112hr4310enr/pdf/BILLS-112hr4310enr.pdf

[Wang 2023]
Wang, Quong; Towhidnejad, Massood; Olwell, David; Firley, Dick; & Roedler, Garry. Quality Management. *SEBoK Website.* November 18, 2023. https://sebokwiki.org/wiki/Quality_Management

[Woody 2014]
Woody, Carol; Ellison, Robert; & Nichols, William. *Predicting Software Assurance Using Quality and Reliability Measures.* CMU/SEI-2014-TN-026. Software Engineering Institute, Carnegie Mellon University. 2014. https://insights.sei.cmu.edu/library/predicting-software-assurance-using-quality-and-reliability-measures-2/

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
(Leave Blank)	December 2024	Final

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
Security Engineering Framework (SEF): Managing Security and Resilience Risks Across the Systems Lifecycle	FA8702-15-D-0002

6. AUTHOR(S)

Christopher Alberts, Charles M. Wallen, Carol Woody, Michael Bandor, and Tom Merendino

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213	CMU/SEI-2024-SR-022

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
SEI Administrative Agent AFLCMC/AZS 5 Eglin Street Hanscom AFB, MA 01731-2100	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 a result, software assurance is becoming increasingly important to organizations across all sectors. A key aspect of software assurance is keeping security and resilience risks within an acceptable tolerance across the systems lifecycle. The Security Engineering Framework (SEF) is a collection of software-focused engineering practices for managing security and resilience risks across the systems lifecycle. It provides a roadmap for building security and resilience into software-reliant systems and maintaining the system's security/resilience capabilities during operations and sustainment (O&S). SEF practices help ensure that engineering processes, software, and tools are secure and resilient, reducing the risk that attackers will disrupt program and system information and assets. Acquisition programs can use the SEF to assess their current security/resilience engineering practices and chart a course for improvement, ultimately reducing security/resilience risks in deployed software-reliant systems. The SEF organizes practices into a hierarchy of goals and domains and provides in-depth guidance for all goals and practices. SEF guidance describes the capability represented by each goal and provides an elaboration of each practice in the framework. This report provides a detailed description of the SEF, including its organizing structure, practices, and guidance.

14. SUBJECT TERMS	15. NUMBER OF PAGES
software assurance, security and resilience risks, framework	176

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

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. Z39-18 298-102