NIST Special Publication
NIST SP 800-160v1r1

Engineering Trustworthy Secure Systems

Ron Ross
Mark Winstead
Michael McEvilley

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.800-160v1r1

NIST NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE

NIST Special Publication
NIST SP 800-160v1r1

Engineering Trustworthy Secure Systems

Ron Ross
Computer Security Division
Information Technology Laboratory

Mark Winstead
Michael McEvilley
The MITRE Corporation

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.800-160v1r1

November 2022

U.S. Department of Commerce
Gina M. Raimondo, Secretary

National Institute of Standards and Technology
Laurie E. Locascio, NIST Director and Under Secretary of Commerce for Standards and Technology

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology (NIST), nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

There may be references in this publication to other publications currently under development by NIST in accordance with its assigned statutory responsibilities. The information in this publication, including concepts and methodologies, may be used by federal agencies even before the completion of such companion publications. Thus, until each publication is completed, current requirements, guidelines, and procedures, where they exist, remain operative. For planning and transition purposes, federal agencies may wish to closely follow the development of these new publications by NIST.

Organizations are encouraged to review all draft publications during public comment periods and provide feedback to NIST. Many NIST cybersecurity publications, other than the ones noted above, are available at https://csrc.nist.gov/publications.

Authority

This publication has been developed by NIST in accordance with its statutory responsibilities under the Federal Information Security Modernization Act (FISMA) of 2014, 44 U.S.C. § 3551 et seq., Public Law (P.L.) 113-283. NIST is responsible for developing information security standards and guidelines, including minimum requirements for federal information systems, but such standards and guidelines shall not apply to national security systems without the express approval of appropriate federal officials exercising policy authority over such systems. This guideline is consistent with the requirements of the Office of Management and Budget (OMB) Circular A-130.

NIST Technical Series Policies

Copyright, Fair Use, and Licensing Statements
NIST Technical Series Publication Identifier Syntax

Publication History

Approved by the NIST Editorial Review Board on 2022-11-08
Supersedes NIST SP 800-160 Vol. 1 (Nov. 2016; updated 2018-03-21) https://doi.org/10.6028/NIST.SP.800-160v1

How to Cite this NIST Technical Series Publication:

Ross R, McEvilley M, Winstead M (2022) Engineering Trustworthy Secure Systems. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-160v1r1. https://doi.org/10.6028/NIST.SP.800-160v1r1

Author ORCID iDs

Ron Ross: 0000-0002-1099-9757

Contact Information

security-engineering@nist.gov
National Institute of Standards and Technology
Attn: Computer Security Division, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930

All comments are subject to release under the Freedom of Information Act (FOIA).

Abstract

This publication describes a basis for establishing principles, concepts, activities, and tasks for engineering trustworthy secure systems. Such principles, concepts, activities, and tasks can be effectively applied within systems engineering efforts to foster a common mindset to deliver security for any system, regardless of the system's purpose, type, scope, size, complexity, or the stage of its system life cycle. The intent of this publication is to advance systems engineering in developing trustworthy systems for contested operational environments (generally referred to as *systems security engineering*) and to serve as a basis for developing educational and training programs, professional certifications, and other assessment criteria.

Keywords

assurance; developmental engineering; engineering trades; field engineering; implementation; information security; information security policy; inspection; integration; penetration testing; protection needs; requirements analysis; resilience; review; risk assessment; risk management; risk treatment; security architecture; security design; security requirements; specifications; stakeholders; system of systems; system component; system element; system life cycle; systems; systems engineering; systems security engineering; trustworthiness; validation; verification

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. ITL's responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of other than national security-related information in federal information systems. The Special Publication 800-series reports on ITL's research, guidelines, and outreach efforts in information system security, and its collaborative activities with industry, government, and academic organizations.

Patent Disclosure Notice

NOTICE: ITL has requested that holders of patent claims whose use may be required for compliance with the guidance or requirements of this publication disclose such patent claims to ITL. However, holders of patents are not obligated to respond to ITL calls for patents and ITL has not undertaken a patent search in order to identify which, if any, patents may apply to this publication.

As of the date of publication and following call(s) for the identification of patent claims whose use may be required for compliance with the guidance or requirements of this publication, no such patent claims have been identified to ITL.

No representation is made or implied by ITL that licenses are not required to avoid patent infringement in the use of this publication.

Disclaimer

This publication should be used as a supplement to **International Standard ISO/IEC/IEEE 15288** and other supporting international standards. It is recommended that organizations using this publication obtain the appropriate international standards to understand the context of the material in Appendices G through K. Content from ISO/IEC/IEEE 15288 referenced in this publication is used with permission from the Institute of Electrical and Electronics Engineers. It is noted as follows:

The reprinted material has been updated to reflect any changes in the international standard.

Table of Contents

List of Tables

List of Figures

Preface

On May 12, 2021, the President signed an Executive Order (EO) on Improving the Nation's Cybersecurity [1]. The Executive Order stated,

> The United States faces persistent and increasingly sophisticated malicious cyber campaigns that threaten the public sector, the private sector, and ultimately the American people's security and privacy. The Federal Government must improve its efforts to identify, deter, protect against, detect, and respond to these actions and actors.

The Executive Order further described the holistic nature of the cybersecurity challenges confronting the Nation with computing technology embedded in every type of system from general-purpose computing systems that support businesses to cyber-physical systems that control the operations in power plants and provide electricity to the American people. The Federal Government must bring the full scope of its authorities and resources to bear in order to protect and secure its computer systems, whether the systems are cloud-based, on-premises, or hybrid. The scope of protection and security must include systems that process data (i.e., information technology [IT]) and systems that run the machinery that ensure our safety (i.e., operational technology [OT]).

In response to the EO, there is a need to:

- Identify stakeholder assets and protection needs
- Provide protection commensurate with the significance of asset loss and correlated with threat and adversary capabilities
- Develop scenarios and model the complexity of systems to provide a rigorous basis to reason about, manage, and address the uncertainty associated with that complexity
- Adopt an engineering-based approach that addresses the principles of trustworthy secure design and apply those principles throughout the system life cycle

Building trustworthy, secure systems cannot occur in a vacuum with stovepipes for software, hardware, information technology, and the human element (e.g., designers, operators, users, attackers of these systems). Rather, it requires a transdisciplinary approach to protection, a determination across all assets where loss could occur, and an understanding of adversity, including how adversaries attack and compromise systems. As such, this publication addresses considerations for the engineering-driven actions necessary to develop defensible and survivable systems, including the components that compose and the services that depend on those systems. The objective is to address security issues from the perspective of stakeholder requirements and protection needs and to use established engineering processes to ensure that such requirements and needs are addressed with appropriate fidelity and rigor across the entire life cycle of the system.

Engineering trustworthy, secure systems is a significant undertaking that requires a substantial investment in the requirements, architecture, and design of systems, components, applications, and networks. A trustworthy system provides compelling evidence to support claims that it meets its requirements to deliver the protection and performance needed by stakeholders. Introducing a

disciplined, structured, and standards-based set of systems security engineering activities and tasks provides an important starting point and forcing function to initiate needed change.

"Providing satisfactory security controls in a computer system is in itself a system design problem. A combination of hardware, software, communications, physical, personnel and administrative-procedural safeguards is required for comprehensive security. In particular, software safeguards alone are not sufficient."

"Security Controls for Computer Systems," (The Ware Report), Rand Corporation
Defense Science Board Task Force on Computer Security, February 1970

"Mission assurance requires systems that behave with predictability and proportionality."

General Michael Hayden
Former Director National Security Agency (NSA) and Central Intelligence Agency (CIA) Syracuse University, October 2009

"In the past, it has been assumed that to show that a system is safe, it is sufficient to provide assurance that the process for identifying the hazards has been as comprehensive as possible, and that each identified hazard has one or more associated controls. While historically this approach has been used reasonably effectively to ensure that known risks are controlled, it has become increasingly apparent that evolution to a more holistic approach is needed as systems become more complex and the cost of designing, building, and operating them become more of an issue."

Preface, National Aeronautics and Space Administration (NASA) System Safety Handbook, Volume 1, November 2011

"This whole economic boom in cybersecurity seems largely to be a consequence of poor engineering."

Carl Landwehr
Communications of the Association for Computing Machinery (ACM), February 2015

"Cybersecurity requires more than government action. Protecting our Nation from malicious cyber actors requires the Federal Government to partner with the private sector. The private sector must adapt to the continuously changing threat environment, ensure its products are built and operate securely, and partner with the Federal Government to foster a more secure cyberspace…Incremental improvements will not give us the security we need; instead, the Federal Government needs to make bold changes and significant investments in order to defend the vital institutions that underpin the American way of life."

Executive Order (EO) on Improving the Nation's Cybersecurity, May 2021

"[Systems] security engineering must be fundamental to systems engineering, not just a specialty discipline. Security concepts must be fundamental to [an] engineering education, and security proficiency must be fundamental in development teams. Security fundamentals must be clearly understood by stakeholders and effectively evaluated in a way that considers broad goals with security functions and outcomes."

Security in the Future of Systems Engineering (FuSE), a Roadmap of Foundational Concepts, INCOSE International Symposium, July 2021

Acknowledgments

The authors gratefully acknowledge and appreciate the significant contributions from individuals and organizations in the public and private sectors whose constructive comments improved the overall quality, thoroughness, and usefulness of this publication. In particular, we wish to thank Jeff Brewer, Ken Cureton, Jordan Denmark, Rick Dove, Holly Dunlap, Jim Foti, Michael Hankins, Daryl Hild, M. Lee, Tom Llanso, Jimmie McEver, Perri Nejib, Cory Ocker, Daniel Patrick Pereira, Victoria Pillitteri, Greg Ritter, Thom Schoeffling, Theresa Soloway, Nick Stegman, Gary Stoneburner, Gregory Touhill, Isabel Van Wyk, Adam Williams, Drew Wilson, Carol Woody, William Young, and Michael Zisa. The authors also wish to acknowledge members of the International Council for Systems Engineering (INCOSE), including members of the Systems Security Engineering and the Resilient Systems Working Groups, for numerous discussions on the content of the document. Finally, the authors wish to thank the students participating in INCOSE tutorials and MITRE Systems Security Engineering courses whose comments and valuable insights helped to guide and inform many of the proposed changes in this publication.

Historical Contributions

The authors gratefully acknowledge the contributions of Janet Carrier Oren, one of the original authors of NIST Special Publication 800-160, Volume 1. The authors also wish to acknowledge the following organizations and individuals for their historic contributions to this publication:

Organizations: National Security Agency; Naval Postgraduate School; Department of Defense Office of Acquisition, Technology, and Logistics; Department of Homeland Security Science and Technology Office, Cyber Security Division; International Council on Systems Engineering; United States Air Force; Air Force Institute of Technology; Northrop Grumman Corporation; The MITRE Corporation; The Boeing Company; Lockheed Martin Corporation.

Individuals: Beth Abramowitz, Max Allway, Kristen Baldwin, Dawn Beyer, Debora Bodeau, Paul Clark, Keesha Crosby, Judith Dahmann, Kelley Dempsey, Holly Dunlap, Jennifer Fabius, Daniel Faigin, Jeanne Firey, Robin Gandhi, Rich Graubart, Kevin Greene, Richard Hale, Daryl Hild, Kesha Hill, Danny Holtzman, Cynthia Irvine, Brett Johnson, Ken Kepchar, Stephen Khou, Alvi Lim, Logan Mailloux, Dennis Mangsen, Doug Maughn, Rosalie McQuaid, Joseph Merkling, John Miller, Thuy Nguyen, Perri Nejib, Lisa Nordman, Dorian Pappas, Paul Popick, Roger Schell, Thom Schoeffling, Matthew Scholl, Peter Sell, Gary Stoneburner, Glenda Turner, Edward Yakabovicz, and William Young.

Finally, the authors respectfully acknowledge the seminal work in computer security that began in the 1960s. The vision, insights, and dedicated efforts of those early pioneers in computer security serve as the philosophical and technical foundation for the security principles, concepts, methods, and practices employed in this publication to address the critically important problem of engineering trustworthy secure systems.

VIEWING SECURITY FROM THE PROPER PERSPECTIVE

"For the first few decades as a burgeoning discipline, cybersecurity has been dominated by the development of widgets to address some aspect of the problem. Systems have become increasingly complex and interconnected, creating even more attack opportunities, which in turn creates even more opportunities to create defensive widgets that will bring some value in detecting or preventing an aspect of the attack space. Eventually, this becomes a game of whack-a-mole in which a simulated mole pops up from one of many holes and the objective is to whack the mole before it pops back in its hole. The moles represent new attacks, and the holes represent a huge array of potential vulnerabilities – both known and as-yet-undiscovered.

Underlying [the discipline of] engineering is science. Sometimes engineering gets ahead of science, such as in bridge building, where the fundamentals of material science were not well understood. Many bridges were built; many fell down; some stayed up; designs of the ones that stayed up were copied. Eventually, for engineering to advance beyond some point, science must catch up with engineering. The science underlying cybersecurity [and more generally, security] engineering is complex and difficult. On the other hand, there is no time like the present to start, because it is both urgent and important to the future…."

-- O. Sami Saydjari
Engineering Trustworthy Systems McGraw-Hill, August 2018

THE IMPORTANCE OF SCIENCE AND ENGINEERING

When crossing a bridge, we have a reasonable expectation that the bridge will not collapse and will get us to our destination without incident. For bridge builders, the focus is on equilibrium, static and dynamic loads, vibrations, and resonance. The science of physics combines with civil engineering principles and concepts to produce a product that we deem trustworthy, giving us a level of confidence that the bridge is fit-for-purpose.

For system developers, there are also fundamental principles and concepts that can be found in mathematics, computer science, computer and electrical engineering, systems engineering, and software engineering that when properly employed, provide the necessary trustworthiness to engender that same level of confidence. Trustworthy secure systems are achieved by making a significant and substantial investment in strengthening the underlying systems and system components by employing transdisciplinary systems engineering efforts guided and informed by well-defined security requirements and secure architectures and designs. Such efforts have been proven over time to produce sound engineering-based solutions to complex and challenging systems security problems. Only under those circumstances can we build systems that are adequately secure and exhibit a level of trustworthiness that is sufficient for the purpose for which the system was built.

"Scientists study the world as it is, engineers create the world that never has been."

Theodore von Kármán
1962 National Medal of Science Recipient

CRITICAL SYSTEM BEHAVIORS OF THE FUTURE

"To deliver system behavior, the systems engineer must define a group of subsystems and precisely how those subsystems are to interact with each other. It is the subsystems and their interactions which produce the system-level behavior. Many of us recognize a vehicle that can take a 60-degree curve at 200 miles per hour as possessing a valuable system behavior. Would we as quickly recognize safe, private, trusted, and available as system behaviors? These behaviors require the same careful system-level design and trades to achieve optimal solutions as the performance system behavior I mentioned above. And there is a clear need — investors want the system to keep their data private, to be safe, and to be trustworthy so that their control is not compromised by a cyber threat, and to be highly available.

If we systems engineers are willing to recognize these behaviors as system behaviors, then we are accountable for delivering them as part of our job. If we choose to view these behaviors as attributes of the parts of our system but not the system as a whole, then we are likely to consider them as jobs for the "specialty engineers." I've looked back into past behaviors of our system engineering community. What I find are examples of systems engineers giving our 'specialty engineering' colleagues these challenges by way of the requirements-allocation process. I think we have been wrong to do this. Our "specialty" colleagues are likely to take these allocated requirements and focus on building safe, private, trusted, available parts of a system—rather than in delivering safe, private, trusted, and available system behaviors. It is true you can build a safer system by building safe parts. However, you can't build a truly safe system without having safe parts interacting with each other in a safe manner. The same can be said for other system behaviors (private, trusted, available, and so on)."

-- John A. Thomas
President, INCOSE
INCOSE Insight, July 2013.

1. Introduction

Today's systems[1] are inherently complex. The growth in the size, number, and types of components and technologies[2] that compose those systems as well as the system dependencies result in a range of consequences from inconvenience to catastrophic loss due to adversity[3] within the operating environment. Managing the complexity of trustworthy secure systems requires achieving the appropriate level of confidence in the feasibility, correctness-in-concept, philosophy, and design of a system to produce only the intended behaviors and outcomes. This provides the foundation to address stakeholder protection needs and security concerns with sufficient confidence that the system functions only as intended while subjected to different types of adversity and to realistically bound those expectations with respect to constraints and uncertainty. The failure to address complexity and security will leave the Nation susceptible to potentially serious, severe, or catastrophic consequences.

The term *security* is used in this publication to mean freedom from the conditions that can cause a loss of *assets* with unacceptable consequences.[4] Stakeholders must define the scope of security in terms of the assets to which security applies and the consequences against which security is assessed.[5] *Systems engineering* provides a foundation for a disciplined and structured approach to building assured, trustworthy secure systems. As a systems engineering subdiscipline, *systems security engineering* addresses security-relevant considerations intended to produce secure outcomes. The engineering efforts are conducted at the appropriate level of fidelity and rigor needed to achieve *trustworthiness* and assurance objectives.

> Peter Neumann described the concept of trustworthiness in [2] as follows:
>
> "By trustworthiness, we mean simply worthy of being trusted to fulfill whatever critical requirements may be needed for a particular component, subsystem, system, network, application, mission, enterprise, or other entity. Trustworthiness requirements might typically involve (for example) attributes of security, reliability, performance, and survivability under a wide range of potential adversities. Measures of trustworthiness are meaningful only to the extent that (a) the requirements are sufficiently complete and well defined, and (b) can be accurately evaluated."

[1] A *system* is an arrangement of parts or elements that exhibit a behavior or meaning that the individual constituents do not [3]. The elements that compose a system include hardware, software, data, humans, processes, procedures, facilities, materials, and naturally occurring entities [4].

[2] The term *technology* is used in the broadest context in this publication to include computing, communications, and information technologies, as well as any mechanical, hydraulic, pneumatic, or structural components in systems that contain or are enabled by such technologies. This view of technology provides an increased recognition of the digital, computational, and electronic machine-based foundation of modern complex systems and the growing importance of an assured trustworthiness of that foundation in providing the system's functional capability and interaction with its physical machine and human system elements.

[3] The term *adversity* refers to those conditions that can cause asset loss (e.g., threats, attacks, vulnerabilities, hazards, disruptions, and exposures).

[4] The phrasing used in this definition of security is intentional. Ross Anderson noted in [5] that "now that everything's acquiring connectivity, you can't have safety without security, and these ecosystems are emerging." Reflecting on this observation, the security definition was chosen to achieve alignment with a prevailing *safety* definition.

[5] Adapted from [6].

Systems security engineering provides complementary engineering capabilities that extend the concept of trustworthiness to deliver trustworthy secure systems. Trustworthiness is not only about demonstrably meeting a set of requirements. The requirements must also be complete, consistent, and correct. From a security perspective, a trustworthy system meets a set of well-defined requirements, including security requirements. Through evidence and expert judgment, trustworthy secure systems can limit and prevent the effects of modern adversities. Such adversities come in malicious and non-malicious forms and can emanate from a variety of sources, including physical and electronic. Adversities can include attacks from determined and capable adversaries, human errors of omission and commission, accidents and incidents, component faults and failures, abuses and misuses, and natural and human-made disasters.

> "Security is embedded in systems. Rather than two engineering groups designing two systems, one intended to protect the other, systems engineering specifies and designs a single system with security embedded in the system and its components."
>
> **-- An Objective of Security in the Future of Systems Engineering [7]**

1.1. Purpose and Applicability

This publication is intended to:

- Provide a basis for establishing a discipline for systems security engineering as part of systems engineering in terms of its principles, concepts, activities, and tasks
- Foster a common mindset to deliver security for any system, regardless of its purpose, type, scope, size, complexity, or stage of the system life cycle
- Demonstrate how selected systems security engineering principles, concepts, activities, and tasks can be effectively applied to systems engineering activities
- Advance the field of systems security engineering as a discipline that can be applied and studied
- Serve as a basis for the development of educational and training programs, including individual certifications and other professional assessment criteria

The considerations set forth in this publication are applicable to all federal systems other than those systems designated as national security systems as defined in 44 U.S.C., Section 3542.[6] These considerations have been broadly developed from a technical and technical management perspective to complement similar considerations for national security systems and may be used for such systems with the approval of federal officials who exercise policy authority over such systems. State, local, and tribal governments, as well as private sector entities, are encouraged to consider using the material in this publication, as appropriate.

[6] Increasing the trustworthiness of systems is a significant undertaking that requires a substantial investment in the requirements, architecture, design, and development of systems, system components, applications, and networks. The policy in [8] requires federal agencies to implement the systems security engineering principles, concepts, techniques, and system life cycle processes in this publication for all high-value assets.

The applicability statement is not meant to limit the technical and management application of these considerations. That is, the security design principles, concepts, and techniques described in this publication are part of a trustworthy secure design approach as described in Appendix D and can be applied in any of the following cases:

- **Development of a new capability or system**

  The engineering effort includes such activities as concept exploration, preliminary or applied research to refine the concepts and/or feasibility of technologies employed in a new system, and an assessment of alternative solutions. This effort is initiated during the concept and development stages of the system life cycle.

- **Modification of an existing capability or system**
  - *Reactive modifications to fielded systems:* The engineering effort occurs in response to adversity that diminishes or prevents the system from achieving the design intent. This effort can occur during the production, utilization, or support stages of the system life cycle and may be performed concurrently with or independent of day-to-day system operations.
  - *Planned upgrades to fielded systems while continuing to sustain day-to-day operations:* Planned system upgrades may enhance an existing system capability, provide a new capability, or constitute a technology refresh of an existing capability. This effort occurs during the production, utilization, or support stages of the system life cycle.
  - *Planned upgrades to fielded systems that result in new systems:* The engineering effort is conducted as if developing a new system with a system life cycle that is distinct from the life cycle of a fielded system. The upgrades are performed in a development environment that is independent of the fielded system.

- **Evolution of an existing capability or system**

  The engineering effort involves migrating or adapting a system or system implementation from one operational environment or set of operating conditions to another operational environment or set of operating conditions.[7]

- **Retirement of an existing capability or system**

  The engineering effort removes system functions, services, elements, or the entire system from operation and may include the transition of system functions and services to another system. The effort occurs during the retirement stage of the system life cycle and may be conducted while sustaining day-to-day operations.

- **Development of a dedicated, domain-specific, or special-purpose capability or system**
  - *Security-dedicated or security-purposed system:* The engineering effort delivers a system that satisfies a security-dedicated need or provides a security-oriented purpose and does so as a stand-alone system that may monitor or interact with other systems. Such systems

[7] There is a growing need to reuse or leverage system implementation successes within operational environments that are different from how they were originally designed and developed. This type of reuse or reimplementation of systems within other operational environments is more efficient and represents potential advantages in maximizing interoperability between various system implementations. It should be noted that reuse may violate the assumptions used to determine that a system or system component was trustworthy.

can include surveillance systems, physical protection systems, monitoring systems, and security service provisioning systems.

- *High-confidence, dedicated-purpose system:* The engineering effort delivers a system that satisfies the need for real-time vehicular control, industrial or utility processes, weapons, nuclear power plants, and other special-purpose needs. Such systems may include multiple operational states or modes with varying forms of manual, semi-manual, automated, or autonomous modes. These systems have highly deterministic properties, strict timing constraints, functional interlocks, and severe or catastrophic consequences of failure.

- **Development of a system of systems**

  The engineering effort occurs across a set of constituent systems, each with its own stakeholders, primary purpose, and planned evolution. The composition of the constituent systems into a system of systems as noted in [9] produces a capability that would otherwise be difficult or impractical to achieve. This effort can occur across a variety of system of systems from a relatively informal, unplanned system of systems concept and evolution that emerges over time via voluntary participation to a more formal execution with the most formal being a system of systems concept that is directed, structured, planned, and achieved via a centrally managed engineering effort. Any resulting emergent behavior often introduces opportunities and additional challenges for systems security engineering.

1.2. Target Audience

This publication is intended for systems engineers, security engineers, and other engineering professionals. The term *systems security engineer* is used to include systems engineers and security professionals who apply the concepts and principles and perform the activities and tasks described in this publication.[8] This publication can also be used by professionals who perform other system life cycle activities or tasks, including:

- Individuals with security governance, risk management, and oversight responsibilities
- Individuals with security verification, validation, testing, evaluation, auditing, assessment, inspection, and monitoring responsibilities
- Individuals with acquisition, budgeting, and project management responsibilities
- Individuals with operations, maintenance, sustainment, logistics, and support responsibilities
- Providers of technology-related products, systems, or services
- Educators in academic institutions that offer systems engineering, computer engineering, computer science, software engineering, and computer security programs

[8] Systems security engineering activities and tasks can be applied to a mechanism, component, system element, system, system of systems, processes, or organizations. Regardless of the size or complexity of the entity, a transdisciplinary systems engineering team is needed to deliver systems that are trustworthy and that satisfy the protection needs and concerns of stakeholders. The processes are intended to be tailored to facilitate effectiveness.

1.3. How to Use this Publication

This publication is intended to serve as a reference and educational resource for systems engineers, engineering specialties, architects, designers, and any individuals involved in the development of trustworthy secure systems and system components. It is meant to be flexible in its application to meet the diverse needs of organizations. There is no expectation that all of the technical content in this publication will be used as part of a systems engineering effort. Rather, the concepts and principles for trustworthy secure design in Appendices D through F as well as the systems life cycle processes and security-relevant activities and tasks in Appendices G through K can be selectively employed by organizations – relying on the experience and expertise of the engineering teams to determine what is correct for their purposes. Applying the content of this publication enables the achievement of security outcomes that are consistent with the systems engineering perspective on system life cycle processes.

The system life cycle processes described in this publication can take advantage of any system or software development methodology. The processes are equally applicable to waterfall, spiral, DevOps, agile, and other approaches. The processes can be applied recursively, iteratively, concurrently, sequentially, or in parallel and to any system regardless of its size, complexity, purpose, scope, operational environment, or special nature. The full extent of the application of the content in this publication is guided by stakeholder capability needs, protection needs, and concerns with particular attention paid to considerations of cost, schedule, and performance.

1.4. Organization of this Publication

The remainder of this publication is organized as follows:

- Chapter 2 presents an overview of systems engineering and the fundamental concepts associated with engineering trustworthy secure systems. This includes basic concepts that address the structure and types of systems, systems engineering foundations, and the concepts of trust and trustworthiness of systems and system components.
- Chapter 3 describes foundational system security concepts and an engineering perspective to building trustworthy secure systems. This includes the concepts of security and system security, the nature and character of systems, the concepts of assets and asset loss, reasoning about asset loss, defining protection needs, system security viewpoints, demonstrating system security, and an introduction to systems security engineering.
- Chapter 4 provides a systems security engineering framework that includes a problem context, solution context, and trustworthiness context.

The following sections provide additional information to support the engineering of trustworthy secure systems:

- References
- Appendix A: Glossary
- Appendix B: Acronyms
- Appendix C: Security Policy and Requirements

- Appendix D: Trustworthy Secure Design
- Appendix E: Principles for Trustworthy Secure Design
- Appendix F: Trustworthiness and Assurance
- Appendix G: System Life Cycle Processes Overview
- Appendix H: Technical Processes
- Appendix I: Technical Management Processes
- Appendix J: Organizational Project-Enabling Processes
- Appendix K: Agreement Processes
- Appendix L: Change Log

ENGINEERING-DRIVEN SOLUTIONS

The effectiveness of any engineering discipline first requires a thorough understanding of the problem and consideration of all feasible solutions before acting to solve the identified problem. To maximize the effectiveness of systems security engineering, the security requirements for the protection against asset loss must be driven by business, mission, and all other stakeholder asset loss concerns. The security requirements are defined and managed as a well-defined set of engineering requirements and cannot be addressed independently or after the fact.

In the context of systems security engineering, the term *protection* has a broad scope and is primarily focused on the concept of assets and asset loss resulting in unacceptable consequences. The protection capability provided by a system goes beyond prevention and aims to control the events, conditions, and consequences that constitute asset loss. It is achieved in the form of the specific capability and constraints on system architecture, design, function, implementation, construction, selection of technology, methods, and tools and must be "engineered in" as part of the system life cycle process.

Understanding stakeholder asset protection needs (including assets that they own and assets that they do not own but must protect) and expressing those needs through a set of well-defined security requirements is an investment in the organization's mission and business success in the modern age of global commerce, powerful computing systems, and network connectivity.

2. Systems Engineering Overview

This chapter presents system, systems engineering, trust, and trustworthiness concepts that provide the foundation for engineering trustworthy secure systems.

2.1. System Concepts

Many system concepts are important to inform engineering trustworthy secure systems. This includes what constitutes a system, the structure of a system, categories of systems, and the concept of a system of systems.

2.1.1. Systems and System Structure

A *system*[9] is an arrangement of parts or elements that together exhibit a behavior or meaning that the individual constituents do not. The properties of a system (i.e., attributes, qualities, or characteristics) emerge from the system's parts or elements and their individual properties, as well as the relationships and interactions between and among the parts or elements, the system, and its environment [3]. An engineered system is designed or adapted to interact with an anticipated operational environment to achieve one or more intended purposes while complying with applicable constraints [3]. Figure 1 shows the basic structure of a system, including its constituent system elements.[10] [11]

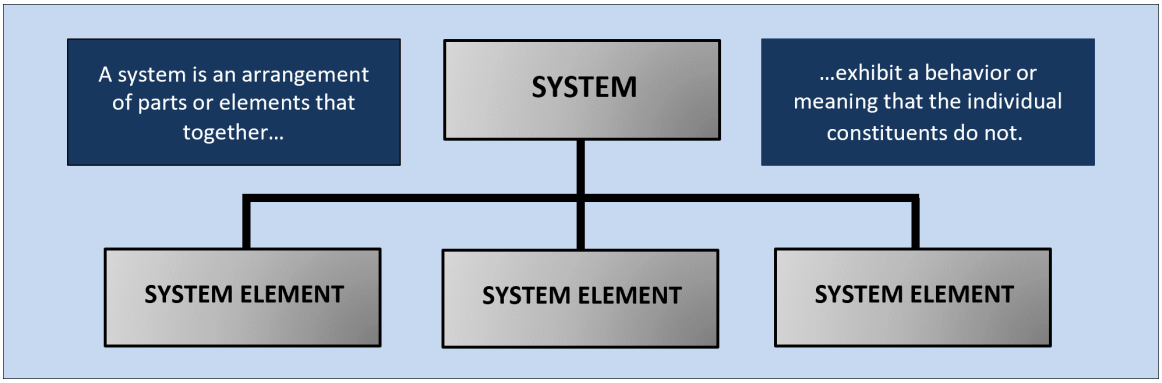

Fig. 1. Basic System and System Element Relationship

The purpose of a system is to deliver one or more capabilities. The capabilities may directly or indirectly interact with, control, or monitor physical, mechanical, hydraulic, or pneumatic devices or other systems or capabilities, or it may provide the ability to create, access, manipulate, transmit, store, or share resources, such as data and information.

[9] Examples of systems include information systems, communications systems, financial systems, manufacturing systems, transportation systems, logistics systems, medical systems, weapons systems, mechanical systems, space systems, industrial control systems (ICS), optical systems, or electrical systems. Systems can be physical or conceptual, use information technology (IT) or operational technology (OT), include humans, be cyber-physical, and leverage Internet of Things (IoT) or other technologies.

[10] A system element can be a discrete component, product, service, subsystem, system, organization, human, infrastructure, or enterprise. System elements are implemented by hardware, software, and firmware that perform operations on information or data; physical structures, devices, and components in the operational environment; and the people, processes, and procedures for operating, sustaining, and supporting the elements.

[11] Systems with few or no active functions (e.g., physical infrastructure) may also exhibit assured trustworthiness. For example, the interstate highway system employs safety barriers such as Jersey walls (a system element) that contribute to the transportation system's trustworthiness.

Figure 2 is a general hierarchical model for the representation of a system. Not all systems, such as networks, are hierarchical in nature. Non-hierarchical systems have models that can more accurately reflect the relationships of their constituent elements. A system element may itself be considered a system (i.e., comprised of other system elements). Realizing a system of interest involves recursively resolving its structure to the point where understandable and manageable system elements can be implemented (i.e., developed, bought, or reused) and subsequently integrating those elements into the system.

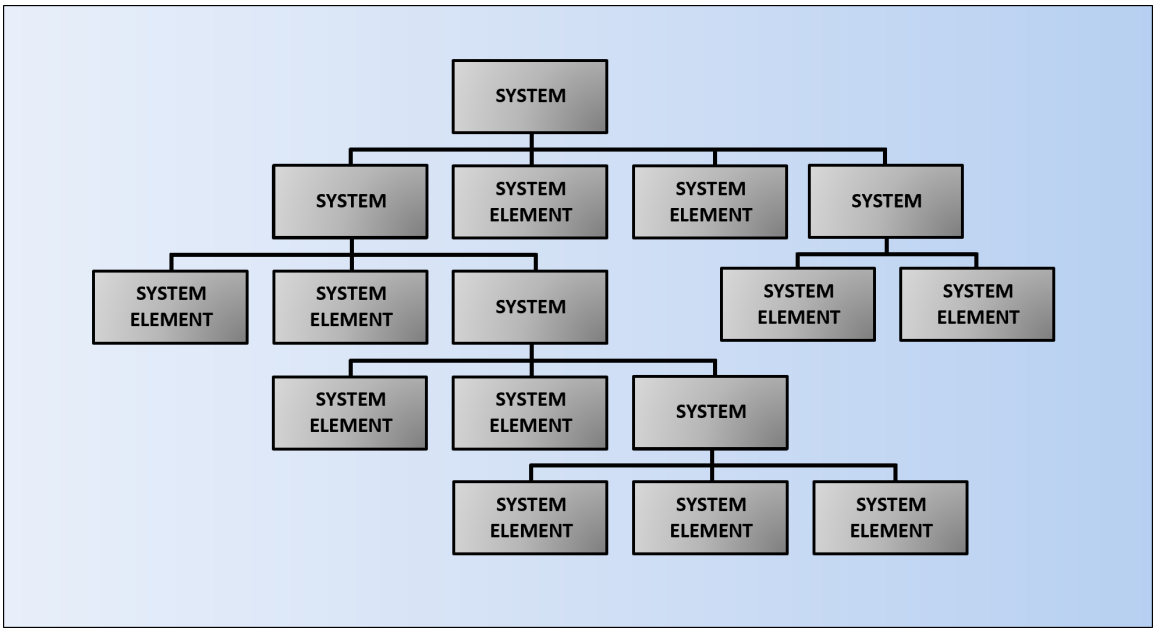

Fig. 2. Model for a System and its Elements

A *system of systems* is a system whose interacting system elements are themselves systems. It provides a unique capability that the constituent systems cannot provide on their own. A system of systems may include inter-system infrastructure, facilities, and processes necessary to enable the constituent systems to integrate or interoperate [10].

2.1.2. Interfacing, Enabling, and Interoperating Systems

Interfacing systems are systems that interact with the system of interest. Interfacing systems have an interface for exchanging data, energy, or other resources with the system of interest. An interfacing system exchanges resources with the system of interest during one or more system life cycle stages, such as a system that interfaces for maintenance purposes or a system used to develop the system of interest. The relationships with interfacing systems can be either bi-directional or one way. Interfacing systems have two specific subsets: enabling systems and interoperating systems.

- *Enabling systems* provide the essential services required to create and sustain the system of interest. Examples of enabling systems include software development environments, production systems, training systems, and maintenance systems.

- *Interoperating systems* interact with the system of interest for the purpose of jointly performing a function.

Figure 3 illustrates the relationship between the system of interest and its interfacing systems in both operational and non-operational (external) environments.

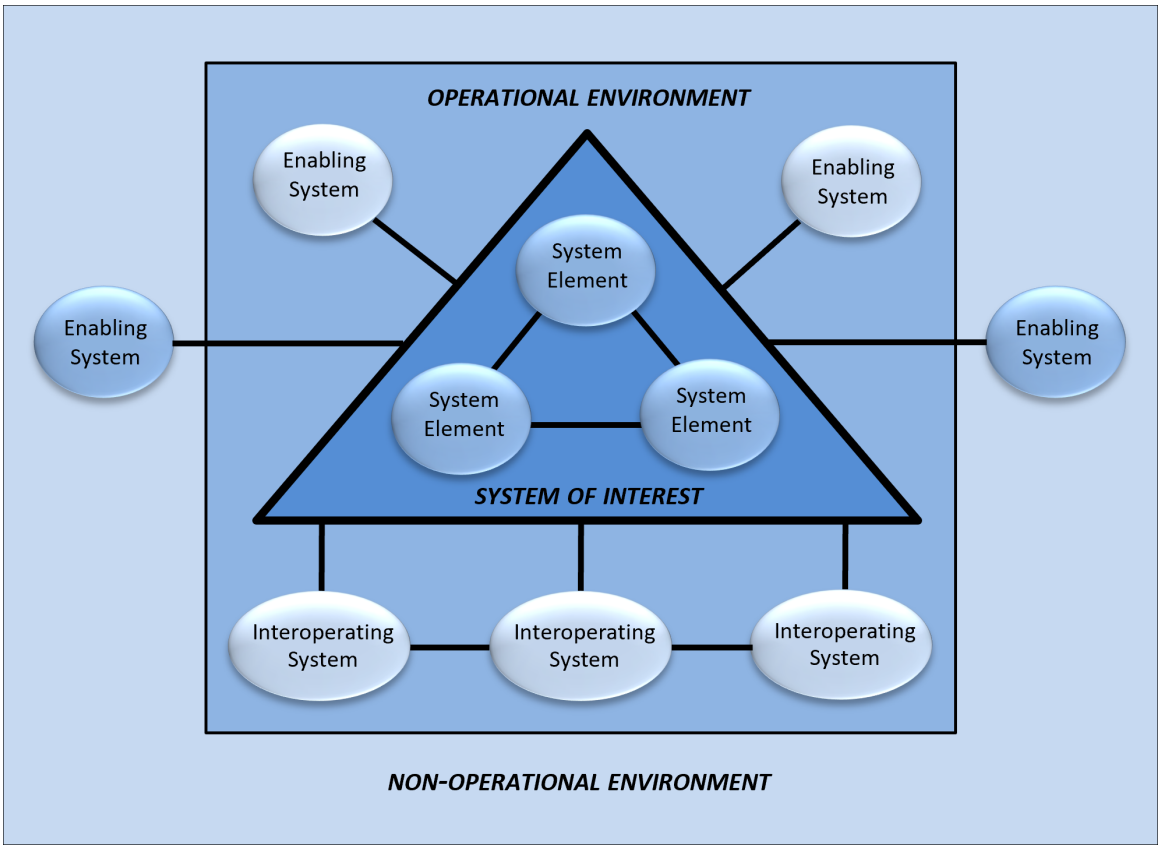

Fig. 3. System of Interest and Interfacing Systems

2.2. Systems Engineering Foundations

Systems engineering is a transdisciplinary[12] and integrative approach to enabling the successful realization, use, and retirement of engineered systems. It employs systems principles and concepts, as well as scientific, technological, and management methods to achieve such systems [12]. Systems engineering is system-holistic in nature, whereby the contributions across multiple engineering and specialty disciplines are evaluated and balanced to produce a coherent system capability. Systems engineering applies systems science and systems thinking[13] to satisfy the often-conflicting needs and priorities of stakeholders within the constraints of cost, schedule,

[12] As noted in [11], transdisciplinary approaches reach "beyond disciplines to find and exploit connections to solve complex problems. Transdisciplinary thinking encourages thinking in terms of new relationships among traditionally distinct disciplines and focusing on new concepts that might arise from such thinking."
[13] Systems science is an interdisciplinary field that studies complex systems in nature, society, and science. It aims to develop interdisciplinary foundations that are applicable in a variety of areas, such as social sciences, engineering, biology, and medicine. Systems thinking is a discipline of examining wholes, interrelationships, and patterns [13].

performance, and effectiveness. The objective is to limit uncertainty and thereby manage risk. Systems engineering is outcome-oriented and leverages engineering processes to realize a system while effectively managing complexity and serving as the principal integrating mechanism for the technical, management, and support activities related to the engineering effort. Finally, systems engineering is data- and analytics-driven to ensure that all decisions and trades are guided and informed by data produced by analyses conducted with an appropriate level of fidelity and rigor.

Systems engineering efforts are complex, system-specific, and context-dependent,[14] requiring close coordination between the *engineering team* and stakeholders throughout the system life cycle stages.[15] While systems engineering is typically considered in terms of its developmental role as part of capability acquisition, systems engineering efforts and responsibilities do not end once a system completes development and is transitioned to the operational environment for day-to-day use. Stakeholders responsible for the system's utilization, support, and retirement provide data to the systems engineering team on an ongoing basis. This data captures the experiences, problems, and issues associated with the operation, maintenance, and sustainment of the system. Stakeholders also advise the engineering team on system enhancements and improvements made or desired. In addition, field engineering provides on-site, full-system life cycle engineering support for operations, maintenance, and sustainment organizations.

There are many additional resources available that provide more in-depth examinations of systems engineering.[16] Such discussions are beyond the scope of this publication.

2.3. Trust and Trustworthiness

The concepts of *trust* and *trustworthiness* are foundational to engineering trustworthy secure systems, to the decisions made to grant trust, and to the extent that trust is granted based on demonstrated trustworthiness. Trust is a belief that an entity meets certain expectations and can be relied upon. The terms *belief* and *can* imply that trust may be granted to an entity whether the entity is trustworthy or not. A trustworthy entity is one for which sufficient evidence exists to support its claimed trustworthiness. Thus, trustworthiness is the demonstrated ability and, therefore, the worthiness of an entity to be trusted to satisfy expectations, including satisfying expectations in the face of adversity. Since trustworthiness is something demonstrated, it is based on evidence that supports a claim or judgment of an entity being worthy of trust [2] [20] [21].

Since trust is not necessarily based on a judgment of trustworthiness, the decision to trust an entity should consider the significance (i.e., consequences, effects, and impacts) of expectations not being fulfilled because of non-performance – whether due to incompetence, deficiency, or

[14] The International Council on Systems Engineering (INCOSE) notes in [14] that "systems engineering in application is specific to stakeholder needs, solution space, resulting system solution(s), and context throughout the system life cycle" and "systems engineering influences and is influenced by internal and external resource, political, economic, social, technological, environmental, and legal factors."

[15] Nomenclature for stages of the system life cycle varies but often includes concept analysis; solution analysis; technology maturation; system design and development; engineering and manufacturing development; production and deployment; training, operations, and support; and retirement and disposal.

[16] INCOSE offers a systems engineering handbook [15] and Systems Engineering Book of Knowledge [13] as general resources. The National Aeronautical and Space Administration (NASA) also offers systems engineering material as it is applied within the NASA community. Publications include the NASA Systems Engineering Handbook [17] and two volumes of expanded systems engineering guidance [18] [19].

failure. Trust that is granted without establishing the required trustworthiness is a significant contributor to risk. The concepts of trust and trustworthiness are discussed in Appendix F.

ENGINEERING FOR TRUST

In January 2022, INCOSE released the Systems Engineering Vision 2035 [16]. It is intended to inspire, guide, and inform the strategic direction for the global systems engineering community. A core element identified for the future state of systems engineering is increased confidence in systems to improve the practice of engineering trusted systems.

As noted in [7], a key problem to address in realizing Vision 2035 is that "systems security has moved from its traditional focus on trust to a more singular focus on risk." The need is to prove a level of system security through evidence-based assurance.

3. System Security Concepts

This chapter describes the aspects necessary for a systems engineering perspective on security. A systems engineering perspective on security requires an understanding of the concept of security (Section 3.1), the concept of an adequately secure system (Section 3.2), and the characteristics of systems (Section 3.3). It also requires an understanding of the concept of assets (Section 3.4), the concepts of loss and loss control (Section 3.5), how to reason about asset loss (Section 3.6), and how to determine protection needs (Section 3.7). In satisfying such needs, specific viewpoints (Section 3.8) and how security is demonstrated are considered, including what is adequate (Section 3.9). The systems engineering subdiscipline that encompasses these considerations is referred to as systems security engineering (Section 3.10).

3.1. The Concept of Security

A system with freedom from those conditions that can cause a loss of assets with unacceptable consequences must provide the intended behaviors and outcomes and also avoid any unintended behaviors and outcomes that constitute a loss. The term *intended* is reflected in two cases, both of which must be satisfied:

- *User intent:* The system behaviors and outcomes expected by the user
- *Design intent:* The system behaviors and outcomes to be achieved by the design

A system that delivers a capability per the design intent but inconsistent with the user intent constitutes a loss. For example, vehicle control loss might result from a failure in the vehicle's steering control function (i.e., failure to meet the design intent) or through an attack that takes control away from the driver (i.e., failure to meet the user intent).

The primary security objective is to ensure that only the intended behaviors and outcomes occur, both with the system and within the system.[17] Every security need and concern derive from this objective, which is based on the concept of *authorization* for what is and is not allowed.[18] As such, the primary security control objective is enforcing constraints in the form of rules for allowed and disallowed behaviors and outcomes. This control objective – and a foundational principle of trustworthy secure design – is *Mediated Access*. If access is not mediated (i.e., controlled though enforcing constraints) following a set of non-conflicting rules, then no basis exists upon which to claim that security is achieved.

The rules for mediated access are stated in a set of security policies[19] that reflect or are derived from laws, directives, regulations, life cycle concepts,[20] requirements, or other specifically stated stakeholder objectives. A security policy includes a scope of control that establishes bounds within which the policy applies. Security policy rules are stated in terms of subjects (active entities), objects (passive entities), and the operations that the subject can perform or invoke on

[17] Intended behaviors include interactions. Relevant interactions include human-to-machine and machine-to-machine interactions. Human-to-machine interactions are transformed into machine-to-machine interactions, whereby a machine element operates on behalf of the human.

[18] An attacker seeks to produce unauthorized behaviors or outcomes. Attackers attempt to accomplish something that they are not authorized to accomplish, even if that behavior or outcome is authorized for some other entity.

[19] A security policy is a set of rules that govern security-relevant system and system element behavior (Appendix C).

[20] Life cycle concepts include operation, sustainment, evolution, maintenance, training, startup, and shutdown.

the object.[21] The rules govern subject-to-object and subject-to-subject behaviors and outcomes. Each security policy rule must be accurate, consistent, compatible, and complete with respect to stakeholder objectives for the defined scope of control.[22] Inconsistency, incompatibility, inaccuracy, or incompleteness in the security policy rules lead to protection gaps. It is equally important that the security protection capabilities of the system are aligned with and can achieve the expectations of the policy.

Privileges[23] define the set of allowed and disallowed behavior and outcomes granted to a subject. Privileges are the basis for making mediated access decisions. A restrictive default practice for security policy enforcement is to design the enforcement mechanism to allow only what the policy explicitly allows and to deny everything else. For a system to be deemed trustworthy secure, there must be sufficient confidence that the system is capable of enforcing the security policy on a continuous basis for the duration of the time that the policy is in effect (Appendix F).

3.2. The Concept of an Adequately Secure System

Adequate security is a concept that enables meaningful judgments about the idealistic nature of security objectives. The definition of security expresses an ideal that encapsulates three essential characteristics of a secure system:

- It enables the delivery of the required system capability despite intentional and unintentional forms of adversity.
- It enforces constraints to ensure that only the desired behaviors and outcomes associated with the required system capability are realized while satisfying the first characteristic.
- It enforces constraints based on a set of rules to ensure that only authorized human-to-machine and machine-to-machine interactions and operations are allowed to occur while satisfying the second characteristic.

These characteristics are to be achieved to the extent practicable, resulting in a gap between the ideal secure system and the security performance that the system can dependably achieve.[24] The judgment that a system is adequately secure[25] requires an evidence-based determination that security performance is optimized against all other performance objectives and constraints. The scope of conditions relevant to security and the acceptable level of security are specific to stakeholder needs. To be adequately secure, the system:

- Meets minimum tolerable levels[26] of security, as determined by experience, analysis, or a combination of both
- Is as secure as reasonably practicable (ASARP)

[21] Active entities exhibit behavior (e.g., a process in execution) while passive entities do not (e.g., data, file).

[22] At the highest level of assurance, security policies are formally specified and verified.

[23] Privileges are also referred to as authorizations or rights.

[24] Because system security is asymmetric – that is, things can be observed to be insecure, but no observation allows one to declare an arbitrary system secure [22] – the ideal cannot be achieved without some uncertainty.

[25] The concept of *adequately secure* is an adaptation of the concept of *adequately safe* from [23].

[26] Below such levels, the system is considered insecure.

As secure as reasonably practicable means that an incremental improvement in security would require a disproportionate deterioration of meeting other system cost, schedule, or performance objectives; would violate system constraints; or would require unacceptable concessions such as an unacceptable change in the way operations are performed.

An adequately secure system does not necessarily preclude all of the conditions that can lead to or result in undesirable consequences. The minimum tolerable levels of security performance and interpretations of *as secure as reasonably practicable* may not be fixed for the life of a system. The information gathered while the system is in use and the lessons learned may guide and inform modifications that raise the bar on either or both (tolerability and practicability).

The concept of adequately secure is, therefore, inherently context-dependent, and subjective in nature. It is based on assertions and expectations about the system security objectives and determining how well those objectives have been achieved. Figure 4 illustrates the trade-offs between system security and the cost, schedule, and technical performance of the system.

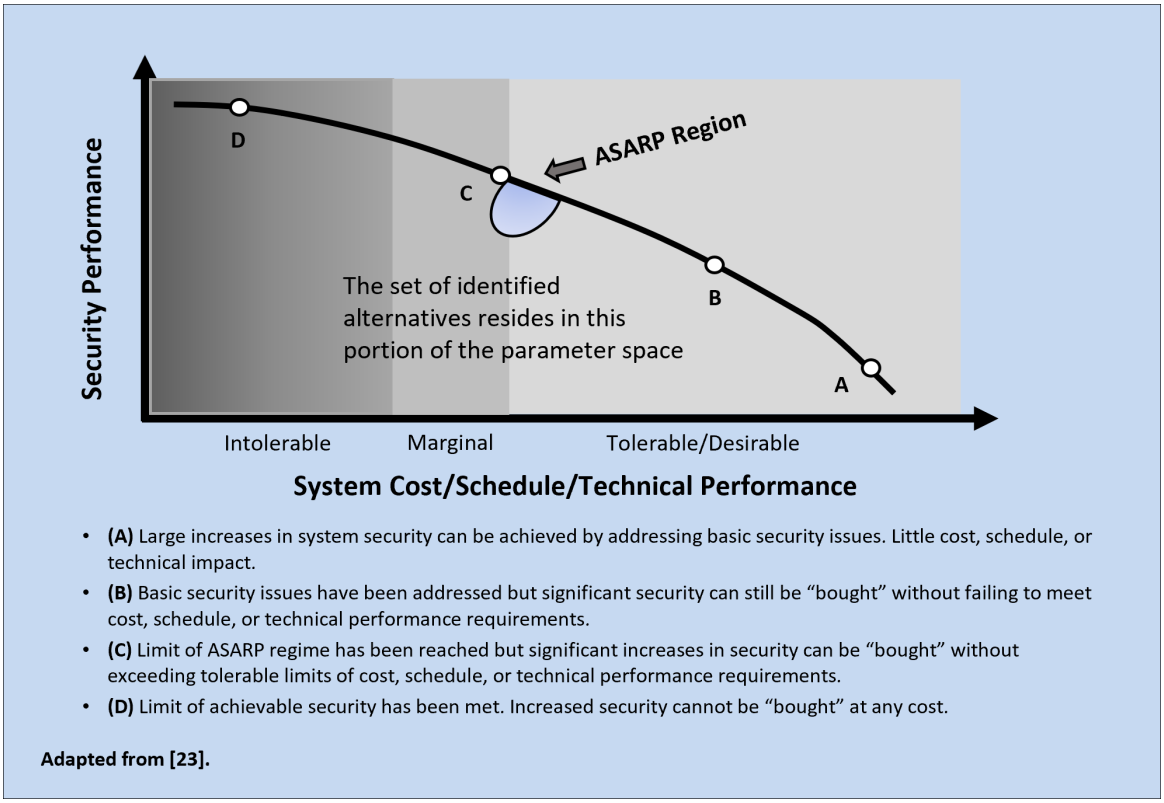

Fig. 4. System Security and Cost/Schedule/Technical Performance

Judging the adequacy of system security requires an understanding of system states. All systems operate in and transition between a set of states. These states and transitions may correspond to or be defined by characteristics of the system, such as how the system functions (e.g., start, run, idle, recovery), how the system is used (e.g., operational, training, maintenance, peacetime, wartime), and by environmental conditions (e.g., under fire or not, temperature ranges). There are security characteristics that determine whether each state or transition is secure, insecure, or indeterminate (i.e., unknown whether secure or insecure). Adequate security depends on being

able to distinguish among secure, insecure, and indeterminate states and to keep the system operating in secure states by applying the principles of *Protective Failure* and *Protective Recovery*.

System states may be secure states (i.e., what states are desired and allowed) and insecure states (i.e., what states are not desired nor allowed). Ideally, a secure system is a system that begins execution in a secure state and does not transition to an insecure state. That is, every state transition results in the same or another secure state. Each state transition must also be secure. Figure 5 illustrates a subset of these idealized secure system state transitions.

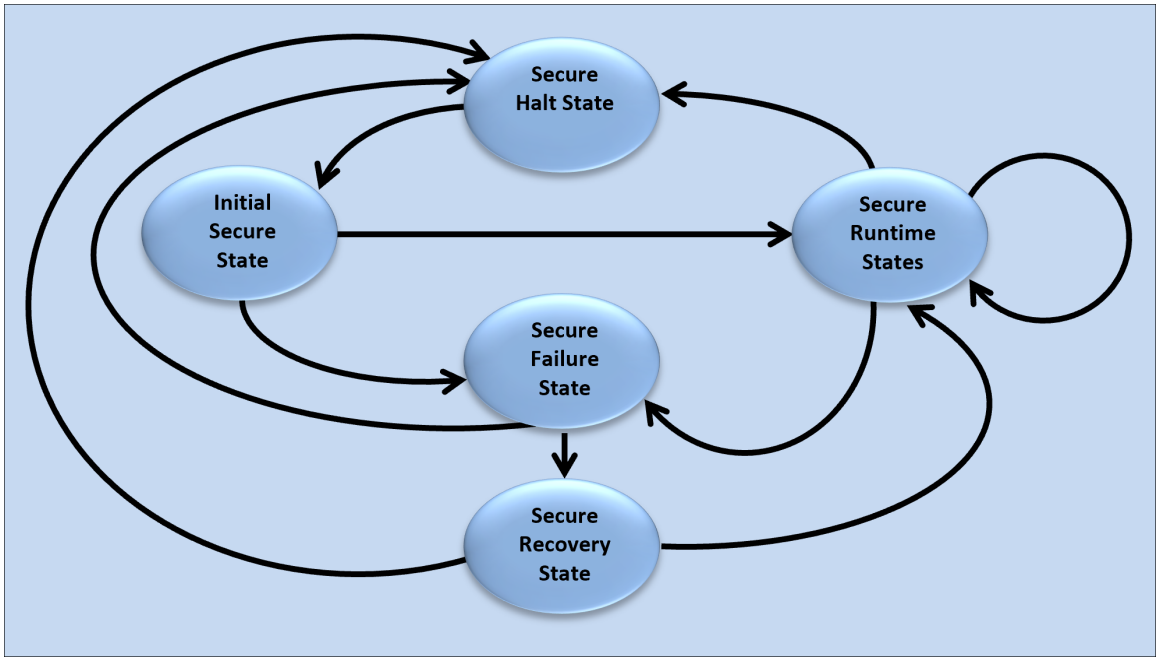

Fig. 5. Idealized Notional Secure System State Transitions

Protective failure requires the ability to (1) detect that the system is in an insecure state and (2) detect a transition that will place the system into an insecure state for the purposes of responding to avoid the propagation of new failure. Protective failure calls for responsive and corrective actions, including (when needed) transitioning to a secure halt state with a protected recovery to allow for the continuation of operations in a reconstituted, reconfigured, or alternative secure operational mode. Other stakeholder objectives may necessitate the continuation of operations in a less-than-fully-secure state and should be reflected as necessary in such things as policy and requirements (Section C.3).

Protective recovery requires the ability to take reactive, responsive, or corrective action to securely transition from an insecure state to a secure state (or a less insecure state). The secure state achieved after completing protective recovery actions includes those actions that limit or prevent any further state transition and those that constitute a type of degraded capability, mode, or operation.

3.3. Characteristics of Systems

The characteristics of systems, their interrelationships with other systems, and their roles within a system of systems all impact security and determinations of adequate security and system trustworthiness. These system characteristics can include:

- System type, function, and primary purpose
- System technological, mechanical, physical, and human element characteristics
- System states and modes of operation
- Criticality or importance of the system
- Ramifications of the system's failure to meet its performance expectations, to function correctly, to produce only the intended behaviors and outcomes, and to provide for its own protection (i.e., self-protection)[27]
- System concept for the delivery of a capability
- Approach to acquisition of the system
- Approach to managerial and operational governance
- Value, sensitivity, and criticality of assets entrusted to and used by the system
- The system's interfaces and the interfacing systems that interact through those interfaces
- Role as a constituent system in one or more system of systems

3.4. The Concept of Assets

An asset is an item of value. There are many different types of assets. Assets are broadly categorized as either tangible or intangible. Tangible assets include physical items, such as hardware, computing platforms, other technology components, and humans. Intangible assets include humans, firmware, software, capabilities, functions, services, trademarks, intellectual property, data, copyrights, patents, image, or reputation.[28] Within asset categories, assets can be further identified and described in terms of common asset classes as illustrated in Table 1.

Table 1. Common Asset Classes

ASSET CLASS	DESCRIPTION	LOSS PROTECTION CRITERIA
MATERIAL RESOURCES AND INFRASTRUCTURE	This asset class includes physical property (e.g., buildings, facilities, equipment) and physical resources (e.g., water, fuel). It also includes the basic physical and organizational structures and facilities (i.e., infrastructure) needed for an activity or the operation of an enterprise or society.[29] An infrastructure may be comprised of assets in other classes. For example, the National Airspace System	*Material resources* are protected from loss if they are not stolen, damaged, or destroyed or are able to function or be used as intended, as needed, and when needed. *Infrastructure* is protected from loss if it meets performance expectations while delivering only the authorized and intended

[27] To the extent feasible, self-protection is a required capability that enables the system to deliver the required stakeholder capabilities while also protecting their assets against loss and the consequences of loss.

[28] Humans are perhaps the most important and valuable of all intangible assets. Safety and security explicitly consider the human asset.

[29] Adapted from the Merriam Webster and Oxford definitions of *infrastructure*.

ASSET CLASS	DESCRIPTION	LOSS PROTECTION CRITERIA
	(NAS) may be considered infrastructure that itself is a system and contains other elements that are forms of systems and infrastructures, such as Air Traffic Control, navigational aids, weather aids, airports, and the aircraft that maneuver within the NAS.	capability and producing only the authorized and intended outcomes.
SYSTEM CAPABILITY	This asset class is the set of capabilities or services provided by the system. Generally, system capability is determined by (1) the nature of the system (e.g., entertainment, vehicular, medical, financial, industrial, or recreational) and (2) the use of the system to achieve mission or business objectives.	*System capability* is protected from loss if the system meets its performance expectations while delivering only the authorized and intended capability and producing only the authorized and intended outcomes.
HUMAN RESOURCES	This asset class includes personnel who are part of the system and are directly or indirectly involved with or affected by the system. The consequences of loss associated with the system may significantly change the importance of this asset class (e.g., the effect on personnel due to a failure of a guidance system in an aircraft is significantly different from the effect on personnel due to the breach of a system that compromises individual credit card information).	*Human resources* are protected from loss if they are not injured, suffer illness, or killed.
INTELLECTUAL PROPERTY[30]	This asset class includes trade secrets, recipes, technology,[31] and other items that constitute an advantage over competitors. The advantage is domain-specific and may be referred to as a competitive advantage, technological advantage, or combative advantage.	*Intellectual property* is protected from loss if it is not stolen, corrupted, destroyed, copied, substituted in an unauthorized manner, or reverse-engineered in an unauthorized manner.
DATA AND INFORMATION	This asset class includes all types of data and information (aggregations of data) and all encodings and representations of data and information (e.g., digital, optical, audio, visual). There are general sensitivity classes of data and information that do not fall within the above categories, such as classified information, Controlled Unclassified Information (CUI), and unclassified data and information.	*Data and information* are protected from loss due to unauthorized alteration, exfiltration, infiltration, and destruction.
DERIVATIVE NON-TANGIBLES	This asset class is comprised of derivative, non-tangible assets, such as image, reputation, and trust. These assets are defined, assessed, and affected – positively and negatively – by the success or failure to provide adequate protection for assets in the other classes.	*Non-tangible assets* are protected from loss by ensuring the adequate protection of assets in the other classes.

Assets may also be considered as individual items or as an aggregate or group of items that spans asset types or asset classes (e.g., personnel data, fire control function, environmental sensor capability). This publication uses the term *asset of interest* to emphasize and establish bounds on the scope of reasoning for a specific asset, asset type, or asset class. The valuation of an asset is a

[30] The term *intellectual property* is defined as an output of a creative human thought process that has some intellectual or informational value [24]. Examples include microcomputer design and computer programs.

[31] The term *technology* is defined as the application of scientific knowledge, tools, techniques, crafts, systems, or methods of organization to solve a problem or achieve an objective [25].

key input in decision-making about investments to protect an asset (Section 3.6). For those cases where an asset is associated with multiple stakeholders, there may be differing, contradictory, competing, or conflicting views about the valuation that must be resolved.

3.5. The Concepts of Loss and Loss Control

Loss is the experience of having an asset taken away or destroyed or the failure to keep or to continue to have an asset in a desired state or form.[32] A loss typically results from an adverse event or condition that causes unacceptable ramifications, consequences, or impacts. A specific loss is determined and assessed independent of the causal events and conditions necessary to produce the loss (i.e., the triggering event, such as an error of omission, or the exploitation event, such as an attack). Examples of resultant adverse events or conditions and their ramifications, impacts, or consequences include:

- *Adverse event or condition:* Data is stolen or inadvertently disclosed on a public website and is no longer solely in the possession of the owner or entities authorized by the owner.
- *Ramification, impact, or consequence:* Market share and competitive advantage is taken away because the data that was lost or stolen provided detailed instructions for a precision machining method that no other company possessed.
- *Adverse event or condition:* A vehicle gets a flat tire, which no longer supports the vehicle weight.
- *Ramification, impact, or consequence:* One cannot drive the vehicle and needs alternate transportation to get to work, the store, or go on vacation.
- *Adverse event or condition:* Confidence in the system of interest operating correctly is lost or questioned.
- *Ramification, impact, or consequence:* Trust in the system and its outputs is lost, whether the loss of confidence is justified or not.

While the loss condition or event is negative relative to the intended norm, the effect of the loss can be either neutral/inconsequential or negative/consequential. For example, a flat tire on a vehicle that is used only for off-road excursion is neutral/inconsequential if no such excursion is planned or affected.

Loss may occur because of a single or combination of intentional or unintentional causes, events, and conditions. These may include the authorized or unauthorized use of the system; intentional acts of disruption or subversion; human and machine faults, errors, and failures; human acts of misuse and abuse; and the by-product of emergence, side effects, and feature interaction. These losses may be inconsequential to the mission or business objectives that the system supports. The objectives may still be achieved despite suffering an immediate or eventual loss that impacts other stakeholder objectives.

The potential to experience loss suggests the need for loss control objectives that serve as the basis for judgments about effectively addressing the prevention and limiting of loss. This

[32] Adapted from the Merriam Webster definition of *loss*.

includes the resultant adverse events and conditions and their ramifications. The loss control objectives also serve as the basis to acquire evidence of assurance that the system as designed, built, used, and sustained will adequately protect against loss while achieving its design intent. The loss control objectives reflect an ideal to preserve the assets' characteristics (i.e., state, condition, form, utility) to the extent practicable despite the potential for those characteristics to be changed. The objectives accept uncertainty in the form of limits to what can be done (i.e., not all losses can be avoided) and limits to the effectiveness of what is done (i.e., anything done has its scope of effectiveness and set of potential failure modes).

Due to uncertainty, it is not possible to guarantee that some form of loss will not occur. There is a need to emphasize protection against the effects of loss, including cascading or ripple events (i.e., the immediate effect of a loss causes some additional unintended or undesired effects or causes additional losses to occur). Thus, holistically protecting against loss and the unintended or undesired effects of loss considers the full spectrum of possible loss across types of losses and loss effects associated with each asset class. This is important considering that all forms of adversity are not knowable. Therefore, it is prudent to ensure that there is focus on the effect to be controlled rather than on the cause when protecting against loss.

The loss control objectives in Table 2 address the possibilities to control the potential for loss and the effects of loss given the limits of certainty, feasibility, and practicality. Collectively, the loss control objectives include the concerns attributed to security and to system survivability, safety, and resilience. Note that satisfying loss control objectives may require trade-offs. Avoiding or limiting the loss of one asset may come at the expense of not avoiding or limiting the loss of another asset, as well as having trade-offs with other objectives (e.g., cost and schedule).

Table 2. Loss Control Objectives

LOSS CONTROL OBJECTIVE	DISCUSSION
LOSS PREVENTION *(Prevent the loss from occurring)*	• This is the case where a loss is totally avoided. That is, despite the presence of adversity: - The system continues to provide *only* the intended behavior and produces *only* the intended outcomes. - The desired properties of the system and assets used by the system are retained. - The assets continue to exist. • Loss avoidance may be achieved by any combination of: - Preventing or removing the event or events that cause the loss - Preventing or removing the condition or conditions that allow the loss to occur - Not suffering an adverse effect despite the events or conditions • Terms such as *avoid, continue, delay, divert, eliminate, harden, prevent, redirect, remove, tolerate,* and *withstand* are typically used to characterize approaches to achieving this objective such that a loss does not occur despite the system being subjected to adversity. • The term *tolerate* refers to the objective of fault/failure tolerance, whereby adversity in the form of faults, errors, and failures is rendered inconsequential and does not alter or prevent the realization of authorized and intended system behavior and outcomes (i.e., the faults, errors, and failures are tolerated).
LOSS LIMITATION	• This covers cases where a loss can or has occurred, and the extent of loss is to be limited. • The extent of loss can be limited in terms of any combination of the following:

LOSS CONTROL OBJECTIVE	DISCUSSION
(Limit the extent of the loss)	- Limited dispersion (e.g., migration, propagation, spreading, ripple, domino, or cascading effects) - Limited duration (e.g., milliseconds, minutes, hours, days) - Limited capacity (e.g., diminished utility, delivery of function, service, or capability) - Limited volume (e.g., bits or bytes of data/information) • Decisions to limit the extent of loss may require prioritizing what constitutes acceptable loss across a set of losses, whereby the objective to limit the loss for one asset requires accepting a loss of some other asset. • The extreme case of loss limitation is to avoid destruction of the asset. • Terms such as *tolerate*, *withstand*, *remove*, *continue*, *constrain*, *stop/halt*, and *restart* fall into this category in the case where the loss occurs and the system can, or enables the ability to, limit the effect of the loss. • Loss recovery and loss delay are two means to limit loss: - *Loss Recovery:* Action is taken by the system or enabled by the system to recover (or allow the recovery of) some or all of its ability to function (i.e., behave, interact, produce outcomes) and to recover assets used by the system (e.g., re-imaging, reloading, or recreating data and information, including software in the system). The restoration of the asset, fully or partially, can limit the dispersion, duration, capacity, or volume of the loss. - *Loss Delay:* The loss event is avoided until the adverse effect is lessened or when a delay enables a more robust response or quicker recovery. • System and environmental conditions may be assumed to result in loss, but measures are taken to limit impacts. • Terms such as *contain*, *recover*, *restore*, *reconstitute*, *reconfigure*, and *restart* are typically used to characterize approaches to achieving this objective.

3.6. Reasoning about Asset Loss

As shown in Figure 6, the elements of a structured approach to reason about asset loss include the (1) context of loss, (2) confidence in addressing loss, (3) significance of loss, (4) addressing loss, and (5) cause of loss. The elements provide an asset-protection basis to determine the objectives for a secure system, optimize the system protection capability, and judge the overall suitability and effectiveness of the implemented protections.[33] The elements are also grouped into two objectives to facilitate reasoning about the asset of interest:

- **Objective 1:** Determine asset protection needs
  - *Context of Loss:* The scope and criteria that bounds reasoning about asset loss
  - *Significance of Loss:* The effect of asset loss (or consequences) based on its valuation[34]
  - *Confidence in Addressing Loss:* The assurance to be achieved based on claims-driven and evidence-based arguments about the effectiveness of what is done to address potential and actual loss

[33] Applying the *asset reasoning* approach works equally to reason about assets in terms of mission (i.e., mission-driven asset reasoning), organization (i.e., organization-driven asset reasoning), and enterprise (i.e., enterprise-driven asset reasoning).

[34] Valuation is a stakeholder determination. Factors that stakeholders may consider include the various costs associated with the asset and the effects of loss. The effects of loss may be short-term (e.g., completing a business transaction) or long-term (e.g., extended loss of capability awaiting replacement of asset).

- **Objective 2:** Satisfy asset protection needs
  - *Cause of Loss:* The events, conditions, or circumstances that describe what has happened before and what can happen in the future that constitute the potential for loss to occur
  - *Addressing Loss:* The various actions taken to exercise control over loss to the extent practicable. The control objectives are to prevent loss from occurring and to limit the extent and duration for those losses that do occur. Limiting loss includes recovery from loss to the extent practicable.

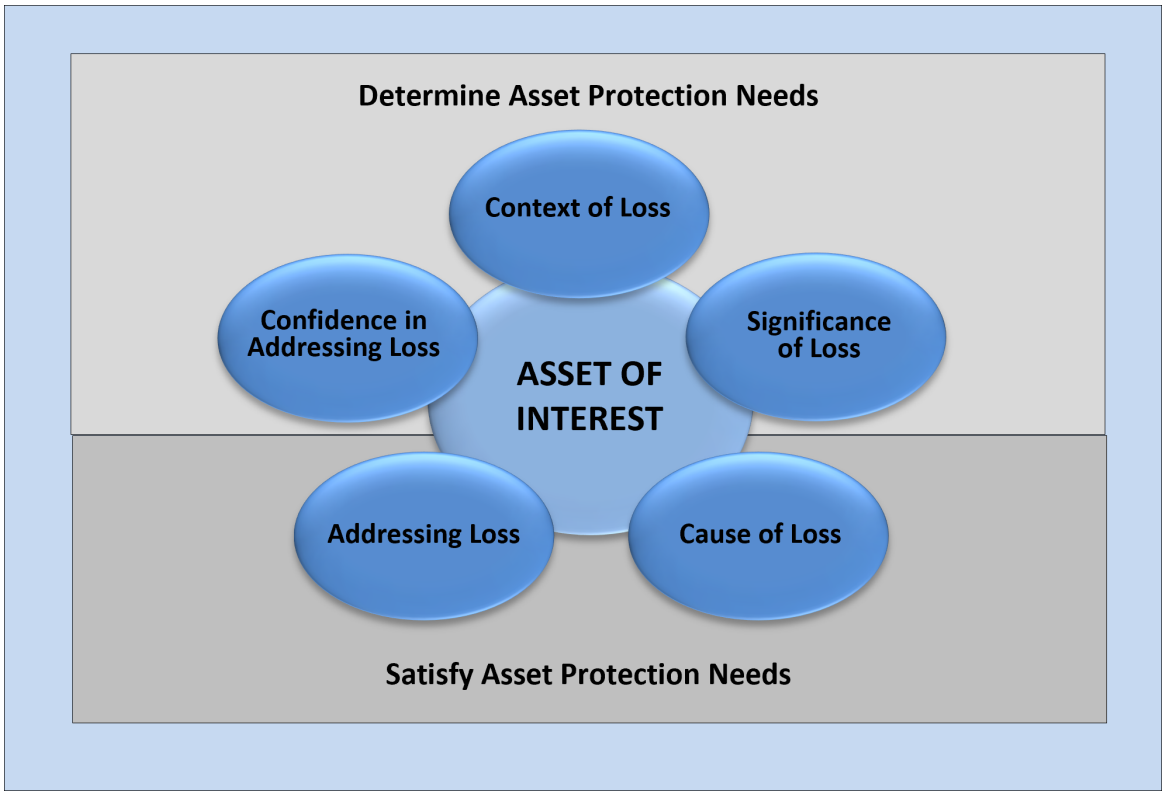

Fig. 6. Reasoning about Asset Protection

The *asset of interest* is the asset class, asset type, or individual asset being addressed. Reasoning about loss is based on the asset of interest. Distinguishing the asset of interest from all other assets provides clarity in the interpretation of loss for the asset of interest and the associated judgments of the suitability and effectiveness of protections employed. A focus on a specific asset class, type, or discrete element also enables precise traceability to requirements that support the analysis needed to determine the protection-relevant impact of changes to requirements.

The *context of loss* establishes the boundary, scope, and time frame for the reasoning, analyses, assessments, and conclusions about the asset of interest. The context of loss also provides a basis to relate and trace asset dependencies and interactions and to group assets for protection. The

context of loss time frame is particularly important because the asset of interest has a life cycle[35] that is different from the system of interest.[36] For example, the asset of interest may be created, configured, or modified outside of the scope of control of the system of interest yet be within the scope of the engineering effort. The asset of interest, once within the scope of control of the system of interest, may have differing protection needs associated with the state or mode of the system (e.g., the system operational mode protection may differ from the system training mode). Additionally, system life cycle assets (Section 3.8) may only exist within a development or production system and their associated supporting environments. The effect of the loss for these assets may transfer to a loss associated with the system of interest. Therefore, the context of loss includes the life cycle of the asset, the state and mode of the system, and other time-based periods or characteristics during which loss is addressed.

TIME FRAME OF LOSS – AN EXAMPLE

A financial portfolio (i.e., an asset or collection of assets) with specific investment objectives and risk acceptance considerations may be created by a financial advisor for a client, funded by the client, and subsequently managed using multiple systems across one or more institutional investment firms throughout the portfolio's life cycle. Each asset of interest within the portfolio may have differing protection needs at different times depending on the type of asset, market conditions, regulatory jurisdiction, risk position, and other asset management factors that are imposed on the system.

The *significance of loss* is the adverse effect (consequence) on the asset of interest or the resultant adverse effect associated with the asset. The significance of loss is best described as an experience that is to be avoided, thereby warranting an investment to protect against the loss occurring and to minimize the extent of the adverse effect should the loss occur. The significance of loss is determined and assessed as an effects-based judgment. That is, it is determined without any consideration of how or why the loss occurs, the probability or likelihood of the loss occurring, and any intent or the absence of intent related to the loss.[37]

The significance of loss answers the following questions:

- What are the ramifications, effects, and problems that result from suffering a loss of the asset of interest?
- What is the severity of those ramifications, effects, and problems?

[35] The lifetime of an asset may be different from the lifetime of the system. Assets may predate the system and may persist after the system's retirement from use. The significance of the loss of an asset can have ramifications that are independent of the system, system function, and business and mission objectives.

[36] The asset life cycle is the same as the system life cycle when the asset of interest is the system of interest. The asset life cycle may be the same or shorter than the system life cycle for those assets created by the system of interest and only required while the system of interest is operating.

[37] Determining the significance of loss is not a determination of risk.

The significance of loss requires clarity in what loss means for the asset of interest. Examples of terms used to describe asset loss include ability, accessibility, accuracy, assurance, advantage (technological, competitive, combatant), capability, control, correctness, existence, investment, ownership, performance, possession, precision, quality, satisfaction, and time.

SIGNIFICANCE OF LOSS – AN EXAMPLE

The significance of loss due to a flat tire is determined and assessed without consideration for how or why the tire became flat (e.g., puncture, manufacturing defect, impact with curb or other object) and without any consideration of malicious intent (e.g., tire cut, valve stem loosened). Regardless of how or why the tire became flat, the significance of loss remains the same (e.g., loss of control if the vehicle is moving, inability to drive if the vehicle is stationary, time lost to replace or repair the tire to make the vehicle operable). The significance of loss due to a flat tire includes the inability to steer the vehicle, and the resultant adverse effect may be to impact some other object (i.e., a crash). The adverse effect of the loss of steering (loss of control) is specific, while the adverse effect of a crash is general (many other circumstances may result in a crash without any loss of the ability to steer the vehicle).

Confidence in addressing loss ensures that protections have a body of objective evidence that demonstrates the effectiveness, sufficiency, and suitability of protective measures to satisfy asset protection needs. Confidence in addressing loss is cumulative. It begins with determining the loss concerns for the asset of interest and continuously builds as those concerns are better understood and addressed across the context of loss, the significance of loss, the causes of loss, and how loss is addressed. The evidence basis that provides confidence is informed by the verification and validation activities that occur throughout the life cycles of the assets and the system, including requirements elicitation and analysis. A key informing element to those activities is to ensure that the results contribute to the confidence sought.

The *cause of loss*[38] is the individual or combination of events, conditions, and circumstances that result in some form of loss of an asset. The causes of asset loss constitute a continuum that includes intentional, unintentional, accidental, incidental, misuse, abuse, error, defect, fault, weakness, and failure events and conditions [26]. This continuum spans all human-based, machine-based, physical-based, and nature-based drivers of loss. The following considerations apply to reasoning about the causes of loss:

- Single events and conditions that alone can produce the loss
- Combinations, sequences, and aggregate events and conditions

[38] Many terms are used to describe the cause of asset loss. Some of these terms are specific to a community of interest or specialty field, while others span communities and specialties. There are also cases where the same term may be used differently across communities and specialty fields (e.g., the term *threat* has varying interpretations across communities, such as physical security, cybersecurity, commerce, law enforcement, industry, military combat operations, and military intelligence). The terms used as a synonym for the cause of asset loss include *attack*, *breach*, *compromise*, *hazard*, *mishap*, *threat*, *violation*, and *vulnerability*.

- Events and conditions that are desirable, intended, and even planned yet produce unanticipated, unforeseen, and unpredictable results
- Cascading and ripple events and conditions

Finally, the causes of asset loss answer the following questions:

- How can loss occur?
- How has loss occurred in the past?

However, determining how loss can occur does not require asking or answering the question, "What is likely or probable to happen?"[39]

Addressing loss occurs through the protective measures that enforce constraints to ensure that only authorized and intended behaviors and outcomes of the system occur. These include:

- Protective measures that are provided by the machine portion of the system (i.e., the system architecture and design, the use of engineered features and devices within the architecture and design)
- Protective measures that are provided by the *human* in the system (i.e., personnel, practices, procedures, the use of tools to support the human as a system element, and the human role in designing and building the machine part of the system)
- Protective measures that are provided by the *physical environment* (i.e., controlled access areas, facility access points, physical monitoring, environmental controls, fire suppression)

The terminology used to describe means and methods includes mechanisms, configurations, controls, safeguards, countermeasures, features, techniques, overrides, practices, procedures, processes, and inhibits. These may be applied in accordance with governing policies, regulations, laws, practices, standards, and techniques.

ASSET-BASED PROTECTION – ENGINEERING FOR SUCCESS

Do not focus on what is likely to happen. Instead, focus on what can happen, and be prepared. That is what systems security engineering means by adopting a preemptive and reactive strategy (Section D.2) in the form of a concept of secure function that addresses the spectrum of asset loss and associated consequences. This means proactively planning and designing to prevent the loss of an asset that you are not willing to accept, to be able to minimize the consequences should such a loss occur, and to be in an informed position to reactively recover from the loss when it does happen.

[39] This point distinguishes analysis of what can happen from a risk assessment that determines probability greater than zero and less than one that the adverse event will happen.

3.7. Determining Protection Needs

Stakeholders need to achieve their mission or business objectives in a secure manner that preserves assets and limits the extent of asset loss. Asset protection must be continuous, thereby making it possible for stakeholders to have a realistic expectation of continuous success in the ability of their systems to support and achieve their objectives.

The scope and expectations for the protection of assets is foundational to achieving the design intent for a trustworthy secure system. Protection needs typically correlate to the severity of consequences associated with the loss of an asset. The protection needs are determined from all needs, concerns, priorities, and constraints to protect and preserve stakeholder and system assets. There are three perspectives for protection needs: (1) the *stakeholder* perspective, (2) the *system* perspective, and (3) the *trades* perspective. Figure 7 illustrates the key input sources used to define protection needs and the outputs derived from the specification of those needs.

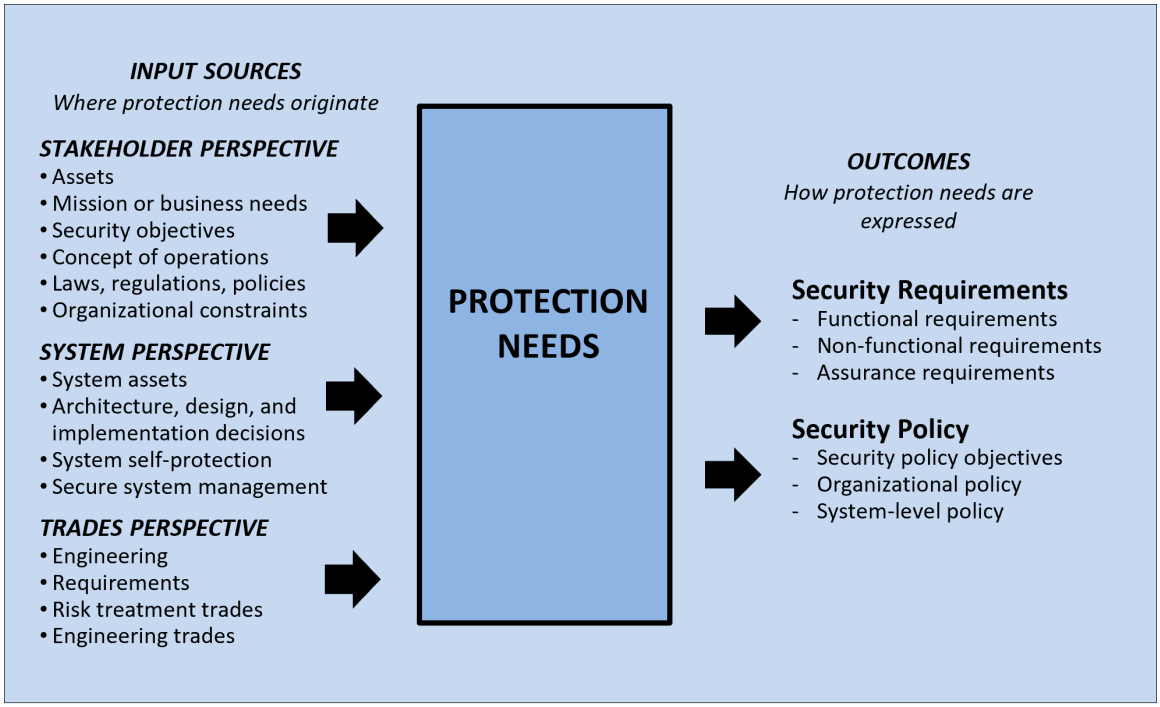

Fig. 7. Defining Protection Needs

The purpose of establishing the need for protection is to decide what assets to protect and to determine the priority given to such protection. This can be accomplished without considering a cause or condition against which to protect. As shown in Figure 8, the need for protection is derived from the relationship among the asset of interest, context of loss, type of loss, and the consequences of loss. This approach establishes the need for protection that – once validated by stakeholders across all assets of interest – provides the basis for developing security objectives and requirements.[40]

[40] Requirements provide a formal and clear expression of the needs, concerns, priorities, and constraints to be satisfied for system function, operation, and maintenance. Each requirement is accompanied by verification methods for demonstrating that the requirement is satisfied. Requirements must be accurate, unambiguous, comprehensive, evaluatable, and achievable.

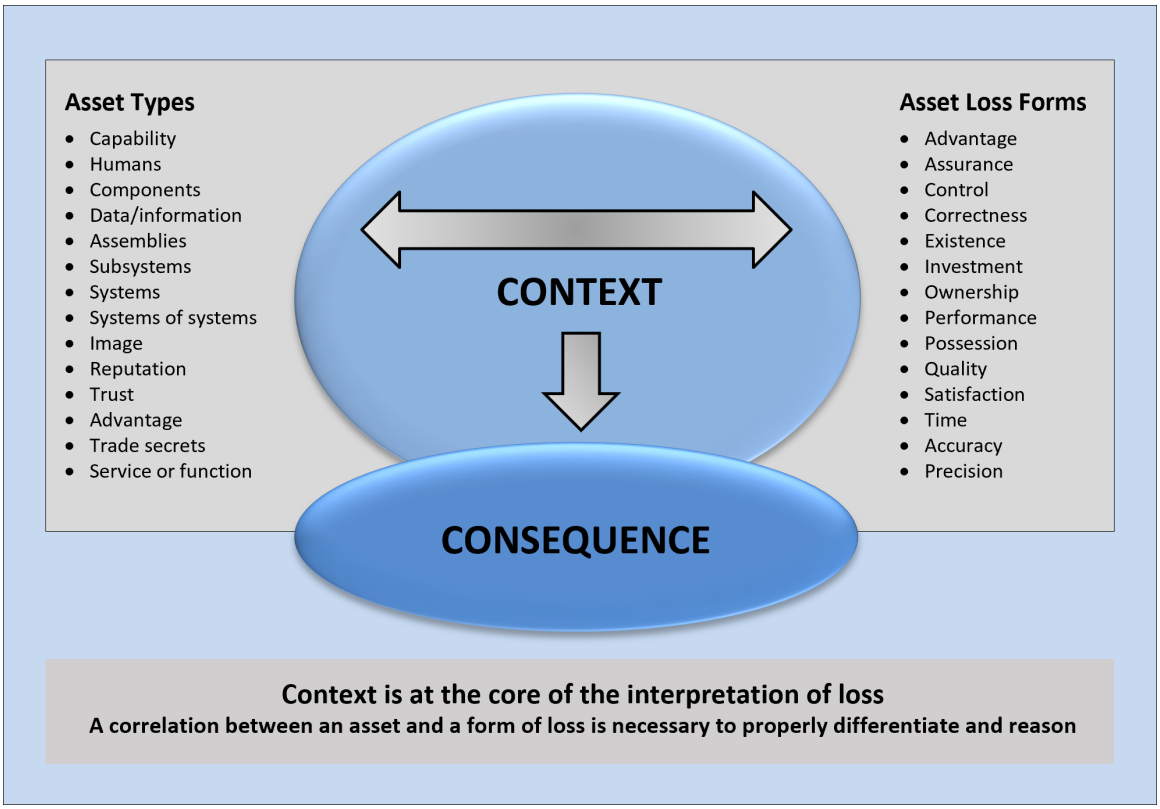

Fig. 8. Relationship among Asset, Loss, and Consequence

To summarize, the following considerations impact the identification of protection needs:

- Assets have different classes and types.
- Assets are associated with stakeholders and the system.
  - Some assets are associated with stakeholders (i.e., stakeholder assets) and have a purpose, use, and existence that is independent of the system being designed.
  - Some assets are associated with the system, are dependent on characteristics of the system design and behavior, and are typically unknown to stakeholders.
- Loss interpretation is dual-faceted.
  - The effect on the asset of interest
  - The effect on those who value the asset of interest
- Loss interpretation is temporal and state-based.
  - Spans a continuum within and across asset types and classes
  - May change across the life cycle of the asset and the state in which the asset exists or is utilized
- Asset-based judgments are subjective.
  - Asset valuation

- Asset loss ramifications
- Asset protection suitability, effectiveness, and dependability

The stakeholder perspective is based on the assets that belong to stakeholders. Therefore, those stakeholders determine the protection needs. The system perspective is based on the assets necessary for the system to function. These assets are determined by system design decisions and the criticality and priority[41] of the asset in providing or supporting the functions of the system. Stakeholders are typically unaware of the existence of system assets and are not able to make decisions about the protection needs for system assets. The protection of system assets is an element of trustworthy secure system design.

Protection needs are continuously reassessed and adjusted as variances, changes, and trades occur throughout the system life cycle. These include the maturation of the system design and life cycle concepts, improved understanding of the operational environment (e.g., a more thorough understanding of adversities), and changes in understanding of the consequences of asset loss. Revisiting protection needs is a necessary part of the iterative nature of systems engineering. Systems security engineering is necessary to ensure completeness in understanding the problem space, exploring all feasible solutions, and engineering a trustworthy secure system.

3.8. System Security Viewpoints

Three predominant viewpoints of system security include *system function*, *security function*, and *life cycle assets*. These viewpoints shape the considerations that are used as trustworthy secure design considerations for any system type, intended use, and consequence of system failure.

Every system is delivered to satisfy stakeholder capability needs. These needs constitute the system function – the system's purpose or role as fulfilled by the totality of the capability it delivers combined with its intended use. The system function is the predominant viewpoint and establishes the context for the security function and the associated system life cycle assets.

The stakeholder capability needs include the protection capability needs. The protection needs parallel the concept of stakeholder capability needs and constitute the system's security function – the totality of the system's purpose or role to securely satisfy stakeholder capability needs. The security function enforces security-driven constraints as part of the overall system design. The purpose of the constraints is to avoid, reduce, and tolerate susceptibilities, defects, weaknesses, and flaws in the system that may constitute vulnerabilities that can be exploited or triggered. These vulnerabilities can reside within the system's structure or behaviors and can have the effect of countering, defeating, or minimizing the ability of the system to effectively satisfy its design intent to deliver the required capability. Thus, the constraints also enable the synthesis of the security function within the system function in a non-conflicting manner.

The security function of the system has both passive and active aspects:

- Passive aspects of the security function do not exhibit behavior (i.e., are non-functional in nature). They include the system architecture and design elements. The passive aspects are

[41] Criticality and priority based on asset valuation are typically used in decisions on protection needs. An asset with higher criticality and priority would take precedence in providing protection should there be constraints that require choosing between the overall protection needs.

part of the system structure and are, therefore, embodied in the architecture of the system. For example, the functional architecture may segment system functions (including security functions) into different subsystems, reducing the possibility of interference among functions as well as limiting the propagation of erroneous behavior. Passive aspects inherently reduce the susceptibility of the system to exposure, hazard, and vulnerability, thereby limiting if not eliminating the potential for loss scenarios. The employment of passive aspects generally enables greater confidence in the protection capability of the system.

- Active aspects of the security function exhibit behavior (i.e., are functional in nature). They include engineered features and devices, referred to as controls, countermeasures, features, inhibits, mechanisms, overrides, safeguards, or services. The active aspects are employed or allocated within the system architecture, have a specific design, and have capabilities and limitations that affect their suitability and effectiveness relative to their intended use.

Passive and active aspects of security function factor into trades, as discussed in Section D.4.4. Active aspects may also require additional hardware or loads on existing hardware; increasing demands for size, weight, and power (SWaP); and making active aspects a challenge for SWaP-restricted systems (e.g., satellites).

Life cycle assets are associated with the system but are not engineered into or delivered with the system. Their association with the system means that they can be the direct cause of loss or a conduit/means through which a loss can occur. Life cycle assets have several types:

- Systems that interact with the system of interest, including conceptual systems
- Intellectual property in various forms, including proprietary algorithms, technologies, and technology solutions
- Data and information associated with the system
- Developmental, manufacturing, fabrication, and production capabilities and systems used to utilize, operate, and sustain the system[42]

3.9. Demonstrating System Security

Demonstrating that a system is adequately secure (Section 3.2) assures stakeholders that their objectives, needs, concerns, and associated constraints have been addressed. Such demonstration must consider the system as an emergent[43] whole that consists of:

- The required capability it delivers
- The protection capability

[42] Examples include software and hardware development tools and suites; modeling and simulation environments and tools; maintenance and diagnostics devices, components, and suites; simulators and test-case scenario generators; and training systems. While these assets are not necessarily within the scope of engineering the system of interest, behaviors and outcomes of these systems have security implications that must be addressed in the secure design of the system of interest. The behaviors and outcomes to consider include how they might directly or indirectly enable, interface, interact, and interoperate with the system of interest.

[43] An *emergent property* is a property exhibited by entities that is meaningful only when attributed to the whole, not to any individual constituent element [27]. Emergent properties of systems include its capability, safety, security, reliability, resilience, survivability, agility, maintainability, and availability. Appendix D discusses emergence in greater detail.

- The limits of certainty[44]

In particular, the limits of certainty apply to requirements and accepting the potential errors, inconsistencies, or gaps in the completeness and coverage of those requirements. Therefore, the requirements and associated verification and validation methods, while a necessary aspect of demonstrating adequate security, are not sufficient to deem a system as adequately secure. The level of confidence provided must be commensurate with the asset loss consequences addressed. The evidence basis for demonstrating confidence must be recorded, traced, maintained, and evolved as variances that are relevant to demonstrating adequate security occur throughout the system life cycle. Additionally, the evidence basis must be meaningful to subject-matter experts across the subjective, competing, and often contradicting needs and beliefs of stakeholders.

Demonstrating this justified confidence or assurance is achieved by an evidentiary basis provided by systems analyses and other evidence-producing activities.[45] The evidentiary basis is used within an approach for structured reasoning, as demonstrated in assurance cases (Section 4.3). The reasoning considers the system needs and capabilities, contributing system quantitative and qualitative factors, and how these capabilities and factors produce an evidentiary base upon which further analyses are conducted in the context of system security. In turn, these analyses support substantiated and reasoned conclusions that serve as the basis for consensus among stakeholders that the system is adequately secure (Appendix F).

> No system can provide **absolute** security due to the limits of human certainty, the uncertainty that exists in the life cycle of every system, and the constraints of cost, schedule, performance, feasibility, and practicality. As such, trade-offs made routinely across contradictory, competing, and conflicting needs and constraints are optimized to achieve **adequate** security, which reflects a decision made by stakeholders.

3.10. Systems Security Engineering

As a subdiscipline of systems engineering, systems security engineering is a transdisciplinary and integrative approach to enabling the successful realization, use, and retirement of engineered trustworthy secure systems. Systems security engineering employs systems, security, and other principles and concepts, as well as scientific, technological, and management methods. Systems security engineering ensures that these principles, concepts, methods, and practices are applied during the system life cycle to achieve stakeholder objectives for assured trustworthiness and asset protection despite adversity. It also helps to reduce and control the causes and conditions that can lead to vulnerability and, as a result, reduces the effect that adversity can have on the system.

[44] An individual function or mechanism can be verified and validated for correctness against its quality and performance attributes. Those results help inform the determination of system security but are insufficient alone.

[45] While the evidence obtained through demonstrating compliance to a set of expectations or criteria may support judgments of adequate security, such evidence alone does not support a claim of adequate security.

Systems security engineering overlaps with other subdisciplines and leverages multiple specialties to accomplish systems security engineering activities and tasks. These specialties include computer security; communications security; transmission security; electronic emissions security; anti-tamper protection; physical security; information, software, hardware, and supply chain assurance; and technology specialties, such as biometrics and cryptography.

Systems security engineering also leverages contributions from other enabling engineering disciplines and specialties to analyze and manage complexity, interconnectedness, dynamicity, and susceptibility associated with hardware, software, and firmware-based technologies.[46] This includes the development, manufacturing, handling, and distribution of technologies throughout the system life cycle.[47]

Figure 9 illustrates the relationships among systems engineering, systems security engineering, and contributing security and other specialty engineering areas.

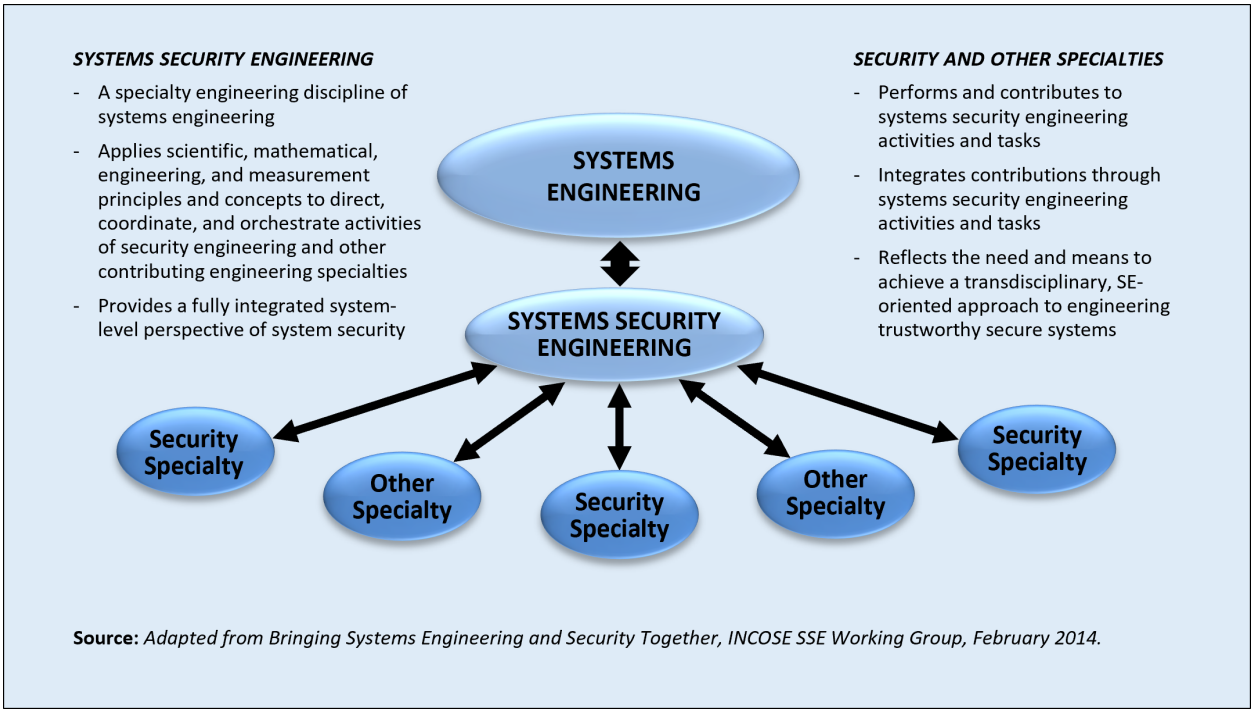

Fig. 9. Systems Engineering and Other Specialty Engineering Disciplines

As part of a transdisciplinary systems engineering effort to deliver a trustworthy secure system, systems security engineering:

- Works with stakeholders to ensure that security objectives, protection needs and concerns, assurance needs, security requirements (including measures of effectiveness [MOEs] and measures of performance [MOPs]), and associated validation methods are defined

[46] Enabling engineering disciplines and specialties include reliability, availability, and maintainability (RAM) engineering; software engineering; resilience engineering; and human factors engineering (ergonomics).

[47] This includes assessing potential supply chain assurance deficiencies when third parties and reuse are considered in planning the system and its realization.

- Defines system security requirements[48] and associated verification methods
- Develops security views and viewpoints of the system architecture and design
- Identifies and assesses susceptibilities and vulnerabilities to life cycle hazards and adversities
- Designs preemptive and reactive features and functions included within a balanced strategy to control asset loss and associated loss consequences
- Provides security considerations to inform systems engineering efforts with the objective to reduce errors, flaws, and weaknesses that may constitute a security vulnerability
- Performs system security analyses and interprets the results of other system analyses in support of decision-making for engineering trades and risk management
- Identifies, quantifies, and evaluates the costs and benefits of security features, functions, and considerations to inform assessments of alternative solutions, engineering trade-offs, and risk treatment[49] decisions
- Demonstrates through evidence-based reasoning that security and trustworthiness claims for the system have been satisfied to the desired level of assurance
- Leverages security and other specialties to address all feasible solutions

SECURITY – AN EMERGING PROPERTY OF AN ENGINEERING PROCESS

A system is engineered to achieve a capability driven by stakeholder mission and business needs. Security is an emergent property of a system that is achieved through a principled engineering process that reflects the stakeholder's protection needs and concerns. The engineered security capability contributes to the overall system capability that satisfies stakeholder mission and business needs. No system can provide absolute security due to the limits of human certainty, the uncertainty that exists in the life cycle of every system, and the constraints of cost, schedule, performance, feasibility, and practicality. As such, trade-offs made routinely across contradictory, competing, and conflicting needs and constraints are optimized to provide adequate security.

[48] It is important to understand the context in which the term *system security requirement* is being used in this publication. For example, due to the complexity of system security, there are several types and purposes of system security requirements (Section 3.8 and Appendix C).

[49] The term *risk treatment* is used in [4] and defined in [28].

4. Systems Security Engineering Framework

The *systems security engineering framework* [29] provides a conceptual view of the key contexts within which systems security engineering activities are conducted. It defines, bounds, and focuses activities and tasks toward achieving stakeholder security objectives and presents a coherent, well-formed, evidence-based case to support judgments about achievement of the objectives.[50] The framework is independent of system type and engineering or acquisition process model. It is not to be interpreted as a sequence of flows or steps but rather as a set of interacting contexts, each with its own checks and balances. The systems security engineering framework emphasizes an integrated, holistic security perspective across all system life cycle stages and is applied to satisfy the milestone objectives of each life cycle stage.

The framework defines three contexts for conducting activities and tasks: (1) the problem context, (2) the solution context, (3) and the trustworthiness context. The three contexts help to ensure that the engineering is driven by a sufficiently complete understanding of the problem. This understanding drives the effort to provide the solution and is supported by a set of activities to design and realize the solution. It also demonstrates the worthiness of the solution in providing adequate security across competing and often conflicting constraints.

While the framework appears to follow a sequential execution across the three contexts, it is intended to be implemented in a closed loop iterative and recursive manner. This approach facilitates a refinement of the problem statement, the proposed solution, and the trustworthiness objectives as the design evolves from concept to the realized solution. The closed loop feedback facilitates interactions among the three framework contexts and the requisite system security analyses to continuously identify and address variances that are introduced into the engineering effort. The feedback loop also helps to achieve continuous process improvement for the system, including viewing the outputs of one life cycle phase (i.e., the solution to the phase) as the inputs to the next phase (i.e., the problem for the next phase).

The three framework contexts share a common foundational base of system security analyses, including system analyses with security interpretations of the analyses results (Section H.6). System security analyses produce data to support engineering and stakeholder decision-making.[51] Such analyses are differentiated for application within the problem, solution, and trustworthiness contexts and employ a variety of concepts, principles, techniques, means, methods, processes, practices, and tools. System security analyses:

- Provide relevant data and technical interpretations of system issues from the system security perspective
- Are differentiated in their application to align with the scope and objectives of where they are applied within the systems security engineering framework

[50] Adapted from [6].

[51] Engineering and stakeholder decision-making involves architecture, assurance, behavior, cost, criticality, design, effectiveness, emergence, exposure, fit-for-purpose, life cycle concepts, penetration resistance, performance (including security performance), protection needs, security objectives, privacy, requirements, resilience, risk, strength of function, threats, trades, uncertainty, vulnerability, verification, and validation.

- Are performed with a level of fidelity, rigor, and formality to produce data with a level of confidence that matches the assurance required by the stakeholders and engineering team (Appendix F)

Figure 10 illustrates the systems security engineering framework and its key components.

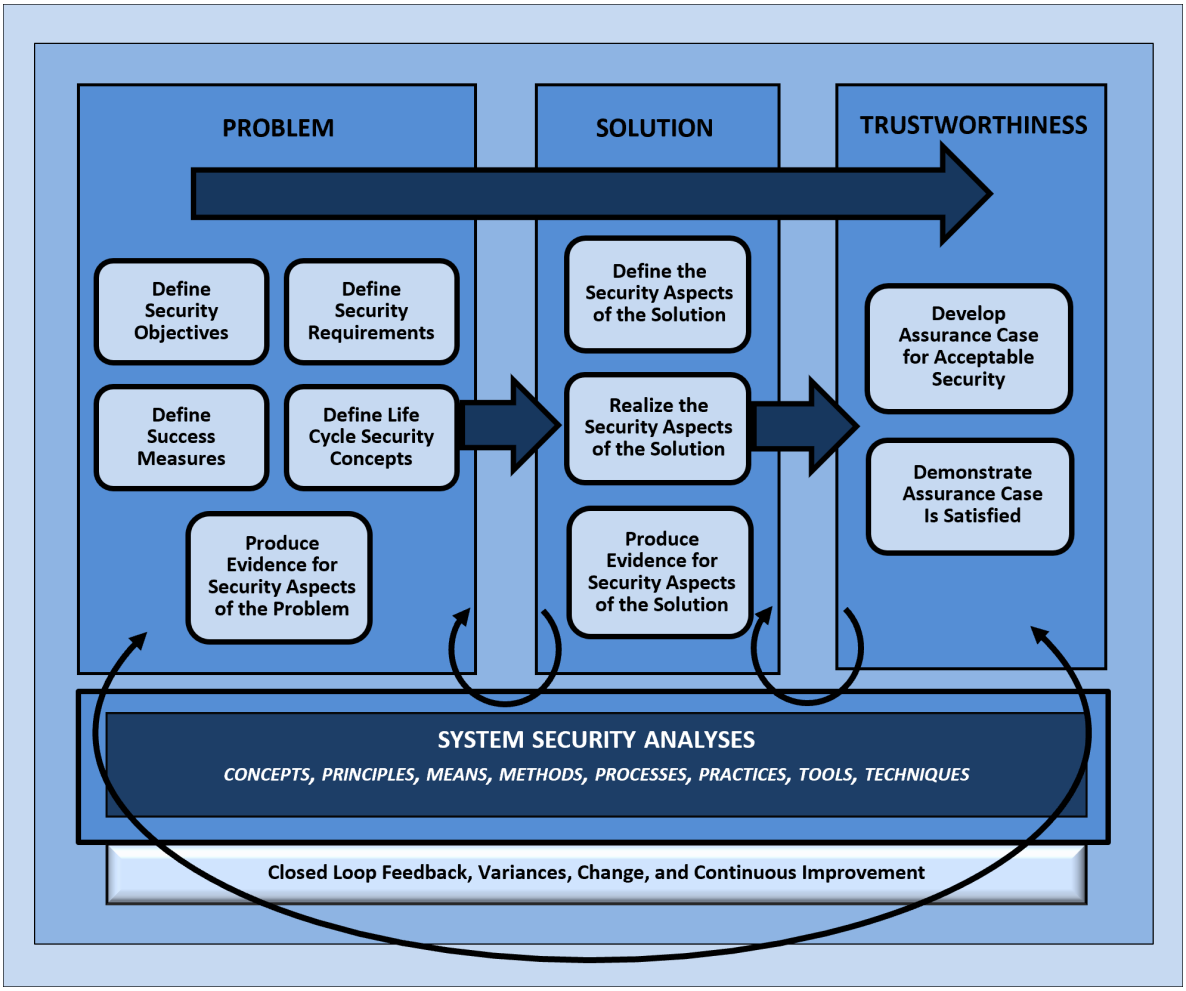

Fig. 10. Systems Security Engineering Framework

4.1. The Problem Context

The problem context defines the basis for an adequately secure system. It focuses on stakeholders' concerns about unacceptable losses given their mission, operational capability, and performance needs and concerns, as well as all associated cost, schedule, performance, and risk-driven constraints. The problem context enables the engineering team to focus on acquiring as complete an understanding of the stakeholder problem as practical, to explore all feasible solution class options, and to select the solution class option or options to be pursued. The problem context includes:

- Defining security objectives

- Defining security requirements
- Determining measures of success
- Determining life cycle security concepts[52]

The security objectives are foundational, establishing and scoping what it means to be adequately secure in terms of protection against asset loss and the significance of such loss. The security objectives have associated measures of success. These measures of success constitute specific and measurable criteria relative to operational performance measures and stakeholder concerns. Measures of success include both the strength of protection and the level of assurance in the protection capability that has been engineered. These measures influence developing security requirements and assurance claims.

Protection needs are determined based on the security objectives, life cycle concepts, and stakeholder concerns. The protection needs are subsequently transformed into stakeholder security requirements and associated constraints, as well as the measures needed to validate that all requirements have been met. A well-defined and stakeholder-validated problem definition and context provide the foundation for all systems engineering and systems security engineering and supporting activities.

The problem context may be interpreted within a life cycle phase as being informed by solutions from earlier life cycle stages, thereby providing a more accurate statement of the problem and its associated constraints. For example, the stakeholder requirements may be the *solution* of an early life cycle phase, which then constrains activities completed in later life cycle stages.

4.2. The Solution Context

The solution context establishes the security aspects and constraints for the architecture and design of the system that (1) satisfies the requirements and objectives of the problem context, (2) realizes the design for the system, and (3) produces sufficient evidence to demonstrate that the requirements and objectives of the problem context have been satisfied.[53] The solution context is based on a balanced preemptive and reactive system security protection strategy[54] that exercises control over events, conditions, asset loss, and the significance of loss to the degree possible, practicable, and acceptable to stakeholders. The solution context includes:

- Defining the security aspects of the solution
- Realizing the security aspects of the solution

[52] The term *life cycle security concept* refers to the processes and activities associated with the system throughout the life cycle (from concept development through retirement) with specific security considerations. It is an extension of the concept of operation and includes the processes and activities related to development, prototyping, assessment of alternative solutions, training, logistics, maintenance, sustainment, evolution, modernization, refurbishment, and disposal. Each life cycle concept has one or more security considerations and constraints that must be fully integrated into the life cycle to ensure that the system security objectives can be met. Life cycle security concepts include those applied during acquisition and program management. Life cycle security concepts can affect such things as Requests for Information, Requests for Proposal, Statements of Work, source selections, development and test environments, operating environments, supply chains, supporting infrastructures, distribution, logistics, maintenance, training, clearances, and background checks.

[53] Security constraints are transformed and incorporated into system design requirements with metadata-tagging to identify security relevance.

[54] The system security protection strategy is consistent with the overall concept of secure function. The concept of secure function, defined during the problem context, constitutes a strategy for a preemptive and reactive protection capability throughout the system life cycle (Section D.2). The strategy has the objective to provide freedom from specific concerns associated with asset loss and loss consequences.

- Producing evidence for the security aspects of the solution

The security aspects of the solution include the development of a system protection strategy; allocated, decomposed, and derived security requirements; security architecture views and viewpoints; security design; security aspects, capabilities, and limitations in the system life cycle procedures; and security performance verification measures. The security aspects of the solution are realized during the implementation of the system design in accordance with the system architecture and in satisfaction of the security requirements. The evidence associated with the security aspects of the solution is obtained with a fidelity and rigor influenced by the level of assurance[55] targeted by the security objectives. Assurance evidence is obtained from standard systems engineering verification methods (e.g., analysis, demonstration, inspection, testing, and evaluation) and complementary validation methods applied against the stakeholder requirements. Application of the solution context may be interpreted to provide a part of the solution, constraining the next iteration of the problem context.

4.3. The Trustworthiness Context

The trustworthiness context is a decision-making context that provides an evidence-based demonstration – through reasoning – that the system of interest is deemed trustworthy (or not) based on a set of claims derived from security objectives. This context consists of:

- Developing and maintaining the assurance case
- Demonstrating that the assurance case is satisfied

The trustworthiness context is grounded in the concept of an assurance case. An assurance case is a well-defined and structured set of arguments and a body of evidence showing that a system satisfies specific claims.[56] Assurance cases provide reasoned, auditable artifacts that support the contention that a top-level claim or set of claims is satisfied, including systematic argumentation and underlying evidence and explicit assumptions that support the claims [30]. The claims may build from subclaims. For a given system life cycle stage, an outcome may sufficiently satisfy a subclaim or set of subclaims, such as a subclaim that stakeholder requirements are sufficiently comprehensive to support a claim that the realized system is adequately secure.

Assurance cases are used to demonstrate that a system exhibits some complex emergent property, such as safety, security, resilience, reliability, or survivability. An effective security assurance case contains foundational security claims derived from security objectives, credible and relevant evidence that substantiates the claims, and valid arguments that relate the various evidence to the supported security claims. The result provides a compelling statement that adequate security has been achieved and driven by stakeholder needs and expectations.

Assurance cases typically include supporting information, such as assumptions, constraints, and inferences that affect the reasoning process. As part of assurance case development, subject-matter expert analyses determine that all security claims are substantiated by the evidence and

[55] Assurance is the measure of confidence associated with a given requirement. As the level of assurance increases, so does the scope, depth, and rigor associated with the methods and analyses conducted (Appendix F).

[56] Software Engineering Institute, Carnegie Mellon University.

the arguments relating the evidence to the claims. Assurance cases must be maintained in response to variances throughout the engineering effort.

The specific form of an assurance case and the level of rigor and formality in acquiring the evidence required is a trade space consideration. It involves the target (i.e., desired) level of assurance, the nature of the consequences for which assurance is sought, and the size and complexity of the dimensions that factor into determining trustworthiness. The assurance case is an engineering construct and must be managed to ensure that the expended effort is justified by the need for the evidence in determining trustworthiness. The assurance claims are the key trustworthiness factor and are developed from the security objectives and associated measures of success independent of the system realization and its supporting evidence. Trustworthiness and assurance are discussed further in Appendix F.

SYSTEMS SECURITY ENGINEERING FRAMEWORK – WHY IT MATTERS

Establishing the problem, solution, and trustworthiness contexts as key components of a systems security engineering framework helps ensure that the security of a system is based on achieving a sufficiently complete understanding of the problem as defined by a set of stakeholder security objectives, security concerns, protection needs, and security requirements. This understanding is essential to developing effective security solutions – that is, a system that is sufficiently trustworthy and adequately secure to protect stakeholder's assets in terms of loss and the associated consequences.

References

[1] Executive Order 14028 (2021), Improving the Nation's Cybersecurity. (The White House, Washington, DC), May 12, 2021. https://www.federalregister.gov/d/2021-10460

[2] Neumann P (2004) Principled Assuredly Trustworthy Composable Architectures, CDRL A001 Final Report, SRI International, Menlo Park, CA. http://www.csl.sri.com/users/neumann/chats4.pdf

[3] Sillitto H, Martin J, McKinney D, Griego R, Dori D, Krob D, Godfrey P, Arnold E, Jackson L (2019) INCOSE-TP-2020-002-06, Systems Engineering and System Definitions. https://www.incose.org/docs/default-source/default-document-library/incose-se-definitions-tp-2020-002-06.pdf?sfvrsn=b1049bc6_0

[4] International Organization for Standardization/International Electrotechnical Commission/Institute of Electrical and Electronics Engineers (2015) ISO/IEC/IEEE 15288:2015 – Systems and software engineering – Systems life cycle processes. https://www.iso.org/standard/63711.html

[5] Anderson R (2020) Security Engineering: A Guide to Building Dependable Distributed Systems (Wiley) 3rd Ed.

[6] National Aeronautics and Space Administration (2011) System Safety Handbook Volume 1: System Safety Framework and Concepts for Implementation, NASA/SP-2010-580, Ver. 1.0. https://ntrs.nasa.gov/api/citations/20120003291/downloads/20120003291.pdf

[7] Dove R, Willett K, McDermott T, Dunlap H, MacNamara DP, Ocker C (2021) Security in the Future of Systems Engineering (FuSE), a Roadmap of Foundational Concepts. INCOSE International Symposium. https://doi.org/10.1002/j.2334-5837.2021.00832.x

[8] Office of Management and Budget (2018) Strengthening the Cybersecurity of Federal Agencies by enhancing the High Value Asset Program. (The White House, Washington, DC), OMB Memorandum M-19-03, December 2018. https://www.whitehouse.gov/wp-content/uploads/2018/12/M-19-03.pdf

[9] Maier M (1998) Architecting Principles for Systems-of-Systems. The Aerospace Corporation. https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291520-6858%281998%291%3A4%3C267%3A%3AAID-SYS3%3E3.0.CO%3B2-D

[10] International Organization for Standardization/International Electrotechnical Commission/Institute of Electrical and Electronics Engineers (2019) ISO/IEC/IEEE 21841:2019 – Systems and software engineering – Taxonomy of systems of systems. https://www.iso.org/standard/71957.html

[11] Madni AM (2017) Transdisciplinary Systems Engineering: Exploiting Convergence in a Hyper-Connected World. Springer International Publishing. https://link.springer.com/book/10.1007/978-3-319-62184-5

[12] International Council On Systems Engineering (2022) What Is Systems Engineering? https://www.incose.org/systems-engineering

[13] BKCASE Editorial Board (2019) The Guide to the Systems Engineering Body of Knowledge (SEBoK), v. 2.0, ed Cloutier RJ (The Trustees of the Stevens Institute of Technology, Hoboken, NJ). https://www.sebokwiki.org/wiki/Guide_to_the_Systems_Engineering_Body_of_Knowledge_(SEBoK)

[14] International Council On Systems Engineering (2022) Systems Engineering Principles. https://www.incose.org/products-and-publications/se-principles

[15] International Council on Systems Engineering (2015) System Engineering Handbook – A Guide for System Engineering Life Cycle Processes and Activities, INCOSE TP-2003-002-04. https://www.incose.org/products-and-publications/se-handbook

[16] International Council on Systems Engineering (2022) Systems Engineering Vision 2035. https://www.incose.org/about-systems-engineering/se-vision-2035

[17] National Aeronautics and Space Administration (2017) Systems Engineering Handbook, NASA SP-2016-6105, Rev 2. https://www.nasa.gov/sites/default/files/atoms/files/nasa_systems_engineering_handbook_0.pdf

[18] National Aeronautics and Space Administration (2016) Expanded Guidance for NASA Systems Engineering. Vol. 1: Systems Engineering Practices. https://ntrs.nasa.gov/citations/20170007238

[19] National Aeronautics and Space Administration (2016) Expanded Guidance for NASA Systems Engineering. Vol. 2: Crosscutting Topics, Special Topics, and Appendices. https://ntrs.nasa.gov/citations/20170007239

[20] Schroeder MD, Clark DD, and Saltzer JH (1977) The Multics Kernel Design Project. Proceedings of Sixth ACM Symposium on Operating Systems Principles. https://web.mit.edu/Saltzer/www/publications/rfc/csr-rfc-140.pdf

[21] Levin T, Irvine C, Benzel T, Bhaskara G, Clark P, and Nguyen T (2007) Design Principles and Guidelines for Security, Technical Report NPS-CS-07-014. Naval Postgraduate School. https://nps.edu/web/c3o/technical-reports

[22] Herley C (2016) Unfalsifiability of Security Claims. Microsoft Research, Proceedings of the National Academy of Sciences. https://doi.org/10.1073/pnas.1517797113

[23] National Aeronautics and Space Administration (2014) System Safety Handbook Vol. 2: System Safety Concepts, Guidelines, and Implementation Examples, NASA/SP-2014-612, Ver. 1.0. https://ntrs.nasa.gov/api/citations/20150015500/downloads/20150015500.pdf

[24] International Organization for Standardization/International Electrotechnical Commission/Institute of Electrical and Electronics Engineers (2017) ISO/IEC/IEEE 24765:2017 – Systems and software engineering – Vocabulary. https://www.iso.org/standard/71952.html

[25] International Organization for Standardization (2013) ISO 16290:2013 – Space systems – Definition of the Technology Readiness Levels (TRLs) and their criteria of assessment. https://www.iso.org/standard/56064.html

[26] Avizienis A, Laprie J, Randell B, and Landwehr C (2004) Basic Concepts and Taxonomy of Dependable and Secure Computing. IEEE Transactions on Dependable and Secure Computing, Vol. 1, No. 1.

[27] International Organization for Standardization/International Electrotechnical Commission/Institute of Electrical and Electronics Engineers (2019) ISO/IEC/IEEE 21840:2019 – Systems and software engineering – Guidelines for the utilization of ISO/IEC/IEEE 15288 in the context of system of systems (SoS). https://www.iso.org/standard/71956.html

[28] International Organization for Standardization (2009) ISO Guide 73:2009 – Risk management – Vocabulary. https://www.iso.org/standard/44651.html

[29] McEvilley M (2015) Towards a Notional Framework for Systems Security Engineering. NDIA 18th Annual Systems Engineering Conference.

[30] International Organization for Standardization/International Electrotechnical Commission (2011) ISO/IEC 15026-2:2011 – Systems and software engineering – Systems and software assurance – Part 2: Assurance case. https://www.iso.org/standard/80625.html

[31] International Organization for Standardization /International Electrotechnical Commission/Institute of Electrical and Electronics Engineers (2018) ISO/IEC/IEEE 29148:2018, Systems and software engineering – Life cycle processes – Requirements engineering. https://www.iso.org/standard/72089.html

[32] International Council on Systems Engineering (2022) Guide to Writing Requirements, Rev. 3.1. https://connect.incose.org/pages/store.aspx

[33] Young W, Leveson NG (2014) An Integrated Approach to Safety and Security based on Systems Theory. Communications of the ACM. Volume 57, Issue 2, 2014, pp. 31-35. https://dl.acm.org/doi/10.1145/2556938

[34] Department of Defense Standard Practice (2012) MIL-STD-882E System Safety.

[35] Uchenick GM, Vanfleet WM (2005) Multiple Independent Levels of Safety and Security: High Assurance Architecture for MSLS/MLS. IEEE Military Communications Conference, pp. 610-614 Vol. 1.

[36] Leveson NG (2011) Engineering a Safer World – Systems Thinking Applied to Safety, Chapter 14, MIT Press, ISBN 978-0-262-01662-9. https://direct.mit.edu/books/book/2908/Engineering-a-Safer-WorldSystems-Thinking-Applied

[37] Anderson J (1972) Computer Security Technology Planning Study, Technical Report ESD-TR-73- 51. Air Force Electronic Systems Division, Hanscom AFB. https://csrc.nist.gov/csrc/media/publications/conference-paper/1998/10/08/proceedings-of-the-21st-nissc-1998/documents/early-cs-papers/ande72a.pdf

[38] Department of Defense (DoD) Standard (1985) 5200.28-STD Trusted Computer System Evaluation Criteria. https://csrc.nist.gov/csrc/media/publications/conference-paper/1998/10/08/proceedings-of-the-21st-nissc-1998/documents/early-cs-papers/dod85.pdf

[39] Hild D, McEvilley M, Winstead M (2021) Principles for Trustworthy Design of Cyber-Physical Systems. MITRE Technical Report, MTR210263. https://figshare.com/articles/preprint/Design_Principles_for_Cyber_Physical_Systems_pdf/15175605

[40] Saleh JH, Marais KB, and Favaro FM (2014) System safety principles: A multidisciplinary engineering perspective. Journal of Loss Prevention in the Process Industries, Vol. 29.

[41] Sheard S, Konrad M, Weinstock C, and Nichols W (2018) A Complexity Measure for System Safety Assurance. INCOSE International Symposium, Adelaide Australia. https://onlinelibrary.wiley.com/doi/abs/10.1002/j.2334-5837.2017.00373.x

[42] Neumann P (2000) Practical Architectures for Survivable Systems and Networks. Technical Report, Final Report, Phase Two, Project 1688, SRI International, Menlo Park, California. http://www.csl.sri.com/neumann/survivability.html

[43] Smith RE (2012) A Contemporary Look at Saltzer and Schroeder's 1975 Design Principles. IEEE Security & Privacy, Vol. 10, No. 6, November/December 2012.

[44] Saltzer JH, Kaashoek MF (2009) Principles of Computer System Design. Morgan Kaufmann Publishers, Burlington, MA.

[45] Moller N, Hansson SO (2008) Principles of Engineering Safety: Risk and Uncertainty Reduction. Reliability Engineering & System Safety, Vol. 93, No. 6, June 2008.

[46] Saltzer JH, Schroeder MD (1975) The Protection of Information in Computer Systems. Proceedings of the IEEE Vol. 63, No. 9, September 1975. https://www.cs.virginia.edu/~evans/cs551/saltzer

[47] Jackson S, Ferris T (2013) Resilience Principles for Engineered Systems. Systems Engineering, Vol. 16, No. 2, July 2013. https://onlinelibrary.wiley.com/doi/abs/10.1002/sys.21228

[48] Simovici DA, Djeraba C (2008) Partially Ordered Sets. Mathematical Tools for Data Mining: Set Theory, Partial Orders, Combinatorics, Springer.

[49] Adcock R, Jackson S, Singer J, Hybertson D (2020) Principles of Systems Thinking. Stevens Institute of Technology. https://www.sebokwiki.org/wiki/Principles_of_Systems_Thinking

[50] Schroeder MD (1972) Cooperation of mutually suspicious subsystems in a computer utility. Ph.D. dissertation, M.I.T., Cambridge, MA. https://web.mit.edu/~saltzer/www/publications/TRs+TMs/Multics/TR-104.pdf

[51] Department of Defense (2007) MIL-HDBK-454B, General Guidelines for Electronic Equipment. https://www.dla.mil/Portals/104/documents/landAndMaritime/v/va/pSMC/documents/lM_MIL_HDBK_454B_151030.pdf

[52] National Security Agency (2002) Technical Report: Information Assurance Technical Framework (IATF), Release 3.1. https://ntrl.ntis.gov/NTRL/dashboard/searchResults/titleDetail/ADA606355.xhtml

[53] Ross R, Pillitteri V, Graubart R, Bodeau D, and McQuaid R (2021) Developing Cyber Resilient Systems: A Systems Security Engineering Approach. (National Institute of Standards and Technology, Gaithersburg, MD), NIST SP 800-160 Volume 2, Revision 1. https://csrc.nist.gov/publications/detail/sp/800-160/vol-2-rev-1/final

[54] Lampson BW (1973) A Note on the Confinement Problem. Communications of the ACM 16, 10, pp. 613-615, October 1973. https://dl.acm.org/doi/10.1145/362375.362389

[55] Popek G (1974) The Principle of Kernel Design. NCC, AFIPS Cong. Proc., Vol. 43.

[56] Benjamin A, et al. (2014) Developing Probabilistic Safety Performance Margins for Unknown and Underappreciated Risks," PSAM-12 International Conf. on Probabilistic Safety and Management, June 2014.

[57] Pagani LP (2004) On the Quantification of Safety Margins. PhD Dissertation, Massachusetts Institute of Technology.

[58] Neumann P (2017) Fundamental Trustworthiness Principles. SRI International, Menlo Park CA.

[59] Bryant WD, Ball RE (2020) Developing the Fundamentals of Aircraft Cyber Combat Survivability: Part 2. Joint Aircraft Survivability Program Office, Aircraft Survivability Journal, Spring 2020.

[60] Ball RE (2003) The Fundamentals of Aircraft Combat Survivability Analysis and Design. 2nd Edition. AIAA Education Series. https://arc.aiaa.org/doi/book/10.2514/4.862519

[61] International Organization for Standardization/International Electrotechnical Commission/Institute of Electrical and Electronics Engineers (2019) ISO/IEC/IEEE 15026-1:2019, Systems and software engineering – Systems and software assurance – Part 1: Concepts and vocabulary. https://www.iso.org/standard/73567.html

[62] Office of the Under Secretary of Defense For Acquisition, Technology, and Logistics, Defense Science Board (2017) Task Force on Cyber Supply Chain. https://dsb.cto.mil/reports/2010s/DSBCyberSupplyChainExecutiveSummary-Distribution_A.pdf

[63] Saydjari OS (2018) Engineering Trustworthy Systems: Get Cybersecurity Design Right the First Time, McGraw-Hill. https://books.apple.com/us/book/engineering-trustworthy-systems-get-cybersecurity-design/id1413527360

[64] Rinehart DJ, Knight JC, and Rowanhill J (2017) Understanding What it Means for Assurance Cases to Work. NASA/CR–2017-219582. https://catalog.libraries.psu.edu/catalog/20766348

[65] Goal Structuring Notation Community Standard, Version 2 (2018) The Assurance Case Working Group. https://scsc.uk/r141B:1?t=1

[66] National Aeronautics and Space Administration (2019) AdvoCATE: Assurance Case Automation Toolset. https://ti.arc.nasa.gov/tech/rse/research/advocate

[67] Joint Task Force (2018) Risk Management Framework for Information Systems and Organizations: A System Life Cycle Approach for Security and Privacy. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-37, Rev. 2. https://doi.org/10.6028/NIST.SP.800-37r2

[68] Joint Task Force (2020) Security and Privacy Controls for Information Systems and Organizations. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-53, Rev. 5. Includes updates as of December 10, 2020. https://doi.org/10.6028/NIST.SP.800-53r5

[69] Snyder D, Powers JD, Bodine-Baron E, Fox B, Kendrick L, Powell MH (2015) Improving the Cybersecurity of U.S. Air Force Military Systems Throughout Their Life Cycles. Rand Corporation. https://www.rand.org/content/dam/rand/pubs/research_reports/RR1000/RR1007/RAND_RR1007.pdf

[70] Department of Defense Instruction 5200.39 (2020) Critical Program Information (CPI) Identification and Protection Within Research, Development, Test, and Evaluation (RDT&E). https://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodi/520039p.pdf

[71] International Organization for Standardization/International Electrotechnical Commission/Institute of Electrical and Electronics Engineers (2011) ISO/IEC/IEEE 42010 – Systems and Software Engineering – Architecture description. https://www.iso.org/standard/50508.html

[72] International Organization for Standardization (2020) ISO 19014:2020 – Earth-moving machinery – Functional safety – Part 4: Design and evaluation of software and data transmission for safety-related parts of the control system. https://www.iso.org/standard/70718.html

[73] International Organization for Standardization (1989) ISO 7498-2:1989 – Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 2: Security Architecture. https://www.iso.org/standard/14256.html

[74] Institute of Electrical and Electronics Engineers (2012) Std. 828-2012 – IEEE Standard for Configuration Management in Systems and Software Engineering, IEEE Computer Society. https://standards.ieee.org/standard/828-2012.html

[75] International Organization for Standardization (1998) ISO 14258:1998 – Industrial automation systems – Concepts and rules for enterprise models. https://www.iso.org/standard/24020.html

[76] International Organization for Standardization (2011) ISO 27026:2011 – Space systems – Programme management – Breakdown of project management structures. https://www.iso.org/standard/43961.html

[77] American National Standards Institute/American Institute of Aeronautics and Astronautics (2018) G-043B-2018, Guide to The Preparation of Operational Concept Documents. https://webstore.ansi.org/Standards/AIAA/ANSIAIAA043B2018

[78] International Organization for Standardization (2015) ISO 9000:2015 – Quality management systems – Fundamentals and vocabulary. https://www.iso.org/standard/45481.html

[79] International Organization for Standardization/International Electrotechnical Commission/Institute of Electrical and Electronics Engineers (2017) ISO/IEC/IEEE 15939:2017 – Systems and software engineering – Measurement process. https://www.iso.org/standard/71197.html

[80] International Organization for Standardization/International Electrotechnical Commission (2009) ISO/IEC 10746-2:2009 – Information technology – Open distributed processing – Reference model: Foundations – Part 2. https://www.iso.org/standard/55723.html

[81] International Organization for Standardization/International Electrotechnical Commission/Institute of Electrical and Electronics Engineers (2021) ISO/IEC/IEEE DIS 24748-6 – Systems and Software Engineering – Life Cycle Management – Part 6: Systems and Software Integration. https://www.iso.org/standard/81563.html

[82] E-Government Act [incl. FISMA] (P.L. 107-347), December 2002. https://www.govinfo.gov/app/details/PLAW-107publ347

[83] International Organization for Standardization (2012) ISO 13008:2012 – Information and documentation – Digital records conversion and migration process. https://www.iso.org/standard/52326.html

[84] International Organization for Standardization/Technical Report (2001) ISO/TR 18307:2001 – Health informatics – Interoperability and compatibility in messaging and communication standards – Key characteristics. https://www.iso.org/standard/33396.html

[85] International Organization for Standardization/International Electrotechnical Commission (2020) ISO/IEC 19989-3:2020(en) – Information security – Criteria and methodology for security evaluation of biometric systems – Part 3, Presentation attack detection. https://www.iso.org/obp/ui/fr/#iso:std:iso-iec:19989:-3:ed-1:v1:en

[86] International Organization for Standardization/International Electrotechnical Commission/Institute of Electrical and Electronics Engineers (2017) ISO/IEC/IEEE

12207:2017 – Systems and software engineering – Software life cycle processes. https://www.iso.org/standard/63712.html

[87] International Organization for Standardization (2019) ISO 17757:2019 – Earth-moving machinery and mining – Autonomous and semi-autonomous machine system safety. https://www.iso.org/standard/76126.html

[88] International Organization for Standardization/International Electrotechnical Commission/Institute of Electrical and Electronics Engineers (2019) ISO/IEC/IEEE 21839:2019 – Systems and software engineering – System of systems (SoS) considerations in life cycle stages of a system. https://www.iso.org/standard/71955.html

[89] Committee on National Security Systems (2022) Instruction No. 4009, Committee on National Security Systems (CNSS) Glossary. https://www.cnss.gov/CNSS/issuances/Instructions.cfm

[90] International Organization for Standardization/International Electrotechnical Commission (2016) ISO/IEC TR 29110-1:2016, Systems and software engineering – Lifecycle profiles for Very Small Entities (VSEs) – Part 1: Overview. https://www.iso.org/standard/62711.html

[91] International Organization for Standardization/International Electrotechnical Commission (2011) ISO/IEC 25010:2011 – Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models. https://www.iso.org/standard/35733.html

[92] International Organization for Standardization/International Electrotechnical Commission/Institute of Electrical and Electronics Engineers (2021) ISO/IEC/IEEE 24774:2021 – Systems and software engineering – Life cycle management – Specification for process description. https://www.iso.org/standard/78981.html

[93] International Organization for Standardization/Society of Automotive Engineers (2021) ISO/SAE 21434:2021 – Road vehicles – Cybersecurity engineering. https://www.iso.org/standard/70918.html

[94] International Organization for Standardization/International Electrotechnical Commission/Institute of Electrical and Electronics Engineers (2018) ISO/IEC/IEEE 24748-2:2018 – Systems and software engineering – Life cycle management – Part 2: Guidelines for the application of ISO/IEC/IEEE 15288 (System life cycle processes). https://www.iso.org/standard/70816.html

[95] Institute of Electrical and Electronics Engineers (2014) IEEE Std 15288.1TM-2014 – Standard for Application of Systems Engineering on Defense Programs, IEEE Computer Society. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7105318

[96] International Organization for Standardization/International Electrotechnical Commission/Institute of Electrical and Electronics Engineers (2018) ISO/IEC/IEEE 24748-1:2018 – Systems and software engineering – Life cycle management – Part 1: Guidelines for life cycle management. https://www.iso.org/standard/72896.html

[97] International Council on Systems Engineering (2022) Needs, Requirements, Verification, Validation Lifecycle Manual. https://connect.incose.org/pages/store.aspx

[98] International Organization for Standardization/International Electrotechnical Commission (2015) ISO/IEC 15026-3:2015 – Systems and software engineering – Systems and software assurance – Part 3: System integrity levels. https://www.iso.org/standard/64842.html

[99] International Organization for Standardization/International Electrotechnical Commission (2021) ISO/IEC 15026-4:2021 – Systems and software engineering – Systems and software assurance – Part 4: Assurance in the life cycle. https://www.iso.org/standard/74396.html

[100] International Organization for Standardization/International Electrotechnical Commission (2008) ISO/IEC 21827:2008 – Information technology – Security techniques – Systems Security Engineering – Capability Maturity Model® (SSE-CMM®). https://www.iso.org/standard/44716.html

[101] International Organization for Standardization/Technical Specification (2010) ISO/TS 18152:2010 – Ergonomics of human-system interaction – Specification for the process assessment of human-system issues. https://www.iso.org/standard/56174.html

[102] International Organization for Standardization/International Electrotechnical Commission (2010) ISO/IEC TR 25060:2010 – Systems and software engineering – Systems and software product Quality Requirements and Evaluation (SQuaRE) – Common Industry Format (CIF) for usability: General framework for usability-related information. https://www.iso.org/standard/35786.html

[103] International Organization for Standardization/International Electrotechnical Commission (2014) ISO/IEC 25063:2014 – Systems and software engineering – Systems and software product Quality Requirements and Evaluation (SQuaRE) – Common Industry Format (CIF) for usability: Context of use description. https://www.iso.org/standard/35789.html

[104] International Organization for Standardization (2010) ISO 9241-210:2010 – Ergonomics of human-system interaction – Part 210: Human-centered design for interactive systems. https://www.iso.org/standard/52075.html

[105] International Organization for Standardization/International Electrotechnical Commission (2019) ISO/IEC 25030:2019 – Software Engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Quality Requirements. https://www.iso.org/standard/72116.html

[106] International Organization for Standardization/International Electrotechnical Commission (2009) ISO/IEC 15408-1:2009 – Information technology – Security techniques – Evaluation criteria for IT security – Part 1: Introduction and general model. https://www.iso.org/standard/72891.html

[107] International Organization for Standardization/International Electrotechnical Commission (2008) ISO/IEC 15408-2:2008 – Information technology – Security techniques – Evaluation criteria for IT security – Part 2: Security functional requirements. https://www.iso.org/standard/72892.html

[108] International Organization for Standardization/International Electrotechnical Commission (2008) ISO/IEC 15408-3:2008 – Information technology – Security techniques – Evaluation criteria for IT security – Part 3: Security assurance requirements. https://www.iso.org/standard/46413.html

[109] International Organization for Standardization/International Electrotechnical Commission (2011) ISO/IEC 27034-1:2011 – Information technology – Security techniques – Application security – Part 1: Overview and concepts. https://www.iso.org/standard/44378.html

[110] International Council on Systems Engineering (2010) Systems Engineering Measurement Primer – INCOSE TP-2010-005-02. https://www.incose.org/docs/default-

source/ProductsPublications/systems-engineering-measurement-primer---december-2010.pdf

[111] International Organization for Standardization/International Electrotechnical Commission (2021) ISO/IEC 27036-1:2021 – Cybersecurity – Supplier relationships – Part 1: Overview and concepts. https://www.iso.org/standard/82905.html

[112] International Organization for Standardization/International Electrotechnical Commission (2022) ISO/IEC 27036-2:2022 – Cybersecurity – Supplier relationships – Part 2: Requirements. https://www.iso.org/standard/82060.html

[113] International Organization for Standardization/International Electrotechnical Commission (2013) ISO/IEC 27036-3:2013 – Information technology – Security techniques – Information security for supplier relationships – Part 3: Guidelines for information and communication technology supply chain security. https://www.iso.org/standard/59688.html

[114] International Organization for Standardization/International Electrotechnical Commission (2015) ISO/IEC 16350:2015 – Information technology – Systems and software engineering – Application management. https://www.iso.org/standard/57922.html

[115] International Organization for Standardization/International Electrotechnical Commission (2006) ISO/IEC 14764:2006 – Software Engineering – Software Life Cycle Processes – Maintenance. https://www.iso.org/standard/39064.html

[116] International Organization for Standardization (2018) ISO 10004:2018 – Quality management – Customer satisfaction – Guidelines for monitoring and managing. https://www.iso.org/standard/71582.html

[117] International Organization for Standardization/International Electrotechnical Commission/Institute of Electrical and Electronics Engineers (2019) ISO/IEC/IEEE 42020:2019 – Software, systems and enterprise – Architecture processes. https://www.iso.org/standard/68982.html

[118] Institute of Electrical and Electronics Engineers (2018) Std. P1012, IEEE Standard for System, Software, and Hardware Verification and Validation, IEEE Computer Society. https://standards.ieee.org/ieee/1012/7324

[119] International Organization for Standardization/International Electrotechnical Commission (2013) ISO/IEC 29119-1:2013 – Software Testing: Concepts and Definitions. https://www.iso.org/standard/45142.html

[120] International Organization for Standardization/International Electrotechnical Commission (2021) ISO/IEC 29119-2:2021 – Software and systems engineering – Software testing – Part 2: Test processes. https://www.iso.org/standard/79428.html

[121] International Organization for Standardization/International Electrotechnical Commission (2021) ISO/IEC 29119-3:2021 – Software and systems engineering – Software testing – Part 3: Test documentation. https://www.iso.org/standard/79429.html

[122] International Organization for Standardization/International Electrotechnical Commission (2021) ISO/IEC 29119-4:2021 – Software and systems engineering – Software testing – Part 4: Test techniques. https://www.iso.org/standard/60245.html

[123] Roedler G, Jones C (2005) Technical Measurement, International Council on Systems Engineering, INCOSE TP-2003-020-01. https://www.incose.org/docs/default-

source/ProductsPublications/technical-measurement-guide---dec-2005.pdf?sfvrsn=4&sfvrsn=4

[124] International Organization for Standardization/International Electrotechnical Commission (2021) ISO/IEC 16085:2021 – Systems and software engineering – Life cycle processes – Risk management. https://www.iso.org/standard/74371.html

[125] International Organization for Standardization (2018) ISO 31000:2018 – Risk management – Guidelines. https://www.iso.org/standard/65694.html

[126] International Organization for Standardization (2017) ISO 10007:2017 – Quality management systems – Guidelines for configuration management. https://www.iso.org/standard/70400.html

[127] Electronic Industries Alliance (2019) EIA 649C, Configuration Management Standard. https://www.sae.org/standards/content/eia649c

[128] International Organization for Standardization/International Electrotechnical Commission/Institute of Electrical and Electronics Engineers (2019) ISO/IEC/IEEE 15289:2019 – Systems and software engineering – Content of life-cycle information items (documentation). https://www.iso.org/standard/74909.html

[129] International Organization for Standardization/International Electrotechnical Commission/Institute of Electrical and Electronics Engineers (2015) ISO/IEC/IEEE 26531:2015 – Systems and software engineering – Content management for product life-cycle, user and service management documentation. https://www.iso.org/standard/43090.html

[130] International Organization for Standardization (2015) ISO 9001:2015 – Quality management systems – Requirements. https://www.iso.org/standard/62085.html

[131] Institute of Electrical and Electronics Engineers (2014) IEEE Std. 730-2014 – IEEE Standard for Software Quality Assurance Processes. https://standards.ieee.org/ieee/730/5284

[132] Department of Defense (2020) DoD Directive 8140.01 – Cyberspace Workforce Management. https://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodd/814001p.PDF?ver=si7QmZONMCW2tStUt4ws3Q%3D%3D

[133] Petersen R, Santos D, Smith MC, Wetzel KA, Witte G (2020) Workforce Framework for Cybersecurity (NICE Framework). (National Institute of Standards and Technology, Gaithersburg, MD), NIST SP 800-181 Revision 1. https://csrc.nist.gov/publications/detail/sp/800-181/rev-1/final

[134] International Organization for Standardization/International Electrotechnical Commission (2015) ISO/IEC 33002:2015 – Information technology – Process assessment – Requirements for performing process assessment. https://www.iso.org/standard/54176.html

[135] Stember, M (1991) Advancing the social sciences through the interdisciplinary enterprise The Social Science Journal, 28:1, 1-14.

Appendix A. Acronyms

ACM
Association for Computing Machinery

AIAA
American Institute of Aeronautics and Astronautics

ANSI
American National Standards Institute

ASARP
As Secure As Reasonably Practicable

CDS
Cross-Domain Solutions

CNSS
Committee on National Security Systems

COTS
Commercial Off-The-Shelf

CUI
Controlled Unclassified Information

DoD
Department of Defense

EIA
Electronic Industries Alliance

EO
Executive Order

FISMA
Federal Information Security Modernization Act

FOIA
Freedom of Information Act

FuSE
Future of Systems Engineering

GSN
Goal Structuring Notation

IEC
International Electrotechnical Commission

IEEE
Institute of Electrical and Electronics Engineers

INCOSE
International Council on Systems Engineering

ISO
International Organization for Standardization

IT
Information Technology

ITL
Information Technology Laboratory

MOE
Measures of Effectiveness

MOP
Measures of Performance

NASA
National Aeronautics and Space Administration

NICE
National Initiative for Cybersecurity Education

NIST
National Institute of Standards and Technology

NDIA
National Defense Industrial Association

OMB
Office of Management and Budget

OT
Operational Technology

SEBoK
Systems Engineering Body of Knowledge

SecDOP
Security Design Order of Precedence

SoS
System of Systems

SP
Special Publication

SSE
Systems Security Engineering

SWaP
Size, Weight, and Power

USC
United States Code

Appendix B. Glossary

abstraction
View of an object that focuses on the information relevant to a particular purpose and ignores the remainder of the information. [24]

acquirer
Stakeholder that acquires or procures a product or service from a supplier. [4]

acquisition
Process of obtaining a system, product, or service. [4]

activity
Set of cohesive tasks of a process. [4]

adequate security (systems)
Meets minimum tolerable levels of security as determined by analysis, experience, or a combination of both and is as secure as reasonably practicable (i.e., incremental improvement in security would require an intolerable or disproportionate deterioration of meeting other system objectives, such as those for system performance, or would violate system constraints).

adverse consequence
An undesirable consequence associated with a loss. [61]

adversity
The conditions that can cause a loss of assets (e.g., threats, attacks, vulnerabilities, hazards, disruptions, and exposures).

agreement
Mutual acknowledgement of terms and conditions under which a working relationship is conducted (e.g., memorandum of agreement or contract). [4]

anomaly
Condition that deviates from expectations based on requirements specifications, design documents, user documents, or standards, or from someone's perceptions or experiences. [24]

anti-tamper
Systems engineering activities intended to prevent or delay exploitation of critical program information in U.S. defense systems in domestic and export configurations to impede countermeasure development, unintended technology transfer, or alteration of a system due to reverse engineering. [70]

See *tampering*.

architecture
Fundamental concepts or properties related to a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution. [71]

See *security architecture*.

architecture (system)
Fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution. [71]

architecture description
A work product used to express an architecture. [71]

architecture framework
Conventions, principles, and practices for the description of architectures established within a specific domain of application and/or community of stakeholders. [71]

architecture view
A work product expressing the architecture of a system from the perspective of specific system concerns. [71]

architecture viewpoint
A work product establishing the conventions for the construction, interpretation, and use of architecture views to frame specific system concerns. [71]

artifact
Work products that are produced and used during a project to capture and convey information (e.g., models, source code). [72]

aspect
The parts, features, and characteristics used to describe, consider, interpret, or assess something.

asset
Anything that has value to a person or organization. [24]

> *Note 1:* Assets have interrelated characteristics that include value, criticality, and the degree to which they are relied upon to achieve organizational mission and business objectives. From these characteristics, appropriate protections are to be engineered into solutions employed by the organization.
>
> *Note 2:* An asset may be tangible (e.g., physical item such as hardware, software, firmware, computing platform, network device, or other technology components) or intangible (e.g., information, data, trademark, copyright, patent, intellectual property, image, or reputation).

assurance
Grounds for justified confidence that a claim has been or will be achieved. [61]

> *Note 1:* Assurance is typically obtained relative to a set of specific claims. The scope and focus of such claims may vary (e.g., security claims, safety claims), and the claims themselves may be interrelated.
>
> *Note 2:* Assurance is obtained through techniques and methods that generate credible evidence to substantiate claims.

assurance case
A reasoned, auditable artifact created that supports the contention that its top-level claim (or set of claims) is satisfied, including systematic argumentation and its underlying evidence and explicit assumptions that support the claim(s). [61]

assurance evidence
The information upon which decisions regarding assurance, trustworthiness, and risk of the solution are substantiated.

> *Note:* Assurance evidence is specific to an agreed-upon set of claims. The security perspective focuses on assurance evidence for security-relevant claims, whereas other engineering disciplines may have their own focus (e.g., safety).

availability
Property of being accessible and usable on demand by an authorized entity. [73]

baseline
Formally approved version of a configuration item, regardless of media, formally designated and fixed at a specific time during the configuration item's life cycle. [74]

> *Note:* The engineering process generates many artifacts that are maintained as a baseline over the course of the engineering effort and after its completion. The configuration control processes of the engineering effort manage baselined artifacts. Examples include stakeholder requirements baseline, system requirements baseline, architecture/design baseline, and configuration baseline.

behavior
The way that an entity functions as an action, reaction, or interaction.

How a system element, system, or system of systems acts, reacts, and interacts. [75]

body of evidence
The totality of evidence used to substantiate trust, trustworthiness, and risk relative to the system.

breakdown structure
Framework for efficiently controlling some aspect of the activities for a program or project. [76]

> *Note:* Examples include work breakdown structure, the decomposition of the defined scope of a project into progressively lower levels consisting of elements of work, and product breakdown structure (i.e., decomposition of a product into its components).

claim
A true-false statement about the limitations on the values of an unambiguously defined property called the claim's property; and limitations on the uncertainty of the property's values falling within these limitations during the claim's duration of applicability under stated conditions. [61]

complex system
A system in which there are non-trivial relationships between cause and effect: each effect may be due to multiple causes; each cause may contribute to multiple effects; causes and effects may be related as feedback loops, both positive and negative; and cause-effect chains are cyclic and highly entangled rather than linear and separable. [3]

component
See *system element*.

concept of operations
Verbal and graphic statement, in broad outline, of an organization's assumptions or intent in regard to an operation or series of operations of new, modified, or existing organizational systems. [77]

> *Note 1:* The concept of operations is frequently embodied in long-range strategic plans and annual operational plans. In the latter case, the concept of operations in the plan covers a series of connected operations to be conducted simultaneously or in succession to achieve an organizational performance objective.
>
> *Note 2:* The concept of operations provides the basis for bounding the operating space, system capabilities, interfaces, and operating environment.

concept of secure function
A strategy for the achievement of secure system function that embodies the preemptive and reactive protection capabilities of the system.

> *Note 1:* This strategy strives to prevent, minimize, or detect the events and conditions that can lead to the loss of an asset and the resultant adverse consequences; prevent, minimize, or detect the loss of an asset or adverse asset consequences; continuously deliver system capability at some acceptable level despite the impact of threats or uncertainty; and recover from adverse asset consequences to restore full system capability or recover to some acceptable level of system capability.
>
> *Note 2:* The concept of secure function is adapted from historical and other secure system concepts, such as the Philosophy of Protection, the Theory of Design and Operation, and the Theory of Compliance.

concern
Matter of interest or importance to a stakeholder. [71]

concern (system)
Interest in a system relevant to one or more of its stakeholders. [71]

configuration item
Item or aggregation of hardware, software, or both that is designated for configuration management and treated as a single entity in the configuration management process. [4]

consequence
Effect (change or non-change), usually associated with an event or condition or with the system and usually allowed, facilitated, caused, prevented, changed, or contributed to by the event, condition, or system. [61]

constraints
Limitation on the system, its design, its implementation, or the process used to develop or modify a system.

Limitation that restricts the design solution, implementation, or execution of the system. [31]

> *Note:* A constraint is a factor that is imposed on the solution by force or compulsion and may limit or modify the design.

control
Purposeful action on or within a process to meet specified objectives.

The mechanism that achieves the action.

criticality
Degree of impact that a requirement, module, error, fault, failure, or other item has on the development or operation of a system.

customer
Organization or person that receives a product. [78]

cyber-physical system
A system integrating computation with physical processes whose behavior is defined by both the computational (digital and other forms) and the physical parts of the system. [27]

data
Representation of facts, concepts, or instructions in a manner suitable for communication, interpretation, or processing by humans or by automatic means.

Collection of values assigned to base measures, derived measures, and/or indicators. [79]

derived requirement
A requirement deduced or inferred from the collection and organization of requirements into a particular system configuration and solution. [31]

> *Note 1:* The next higher-level requirement is referred to as a *parent* requirement, while the derived requirement from this parent is called a *child* requirement.
>
> *Note 2:* A derived requirement is typically identified during the elicitation of stakeholder requirements, requirements analysis, trade studies or validation.

design
Process to define the architecture, system elements, interfaces, and other characteristics of a system or system element. [24]

Result of the process to be consistent with the selected architecture, system elements, interfaces, and other characteristics of a system or system element. [4]

> *Note 1:* Information, including the specification of system elements and their relationships, which is sufficiently complete to support a compliant implementation of the architecture.
>
> *Note 2:* Design provides the detailed implementation-level physical structure, behavior, temporal relationships, and other attributes of system elements.

design characteristics
Design attributes or distinguishing features that pertain to a measurable description of a product or service. [24]

design margin
The margin allocated during design based on assessments of uncertainty and unknowns. This margin is often consumed as the design matures. [17]

domain
A set of elements, data, resources, and functions that share a commonality in combinations of (1) roles supported, (2) rules governing their use, and (3) protection needs. [24]

> *Note:* Security domains may reflect one or any combination of the following: capability, functional, or service distinctions; data flow and control flow associated with capability, functional, or service distinctions; data and information sensitivity; data and information security; or administrative, management, operational, or jurisdictional authority. Security domains that are defined in the context of one or more of these items reflect a protection-focused partitioning of the system that translates to relationships driven by trust concerns.

emergence
The behaviors and outcomes that result from how individual system elements compose to form the system as a whole.

> *Note:* The behavior and outcomes produced by the system are not those of the individual system elements that comprise the system. Rather, the emergent system behavior and outcomes, or properties, result from the composition of multiple system elements.

enabling system
System that supports a system of interest during its life cycle stages but does not necessarily contribute directly to its function during operation. [4]

engineered system
A system designed or adapted to interact with an anticipated operational environment to achieve one or more intended purposes while complying with applicable constraints. [3]

engineering team
The individuals on the systems engineering team with security responsibilities, systems security engineers that are part of the systems engineering team, or a combination thereof.

environment
Context determining the setting and circumstances of all influences upon a system. [71]

error
The difference between desired and actual performance or behavior of a system or system element.

event
Occurrence or change of a particular set of circumstances. [28]

evidence
Grounds for belief or disbelief; data on which to base proof or to establish truth or falsehood.

> *Note 1:* Evidence can be objective or subjective. Evidence is obtained through measurement, the results of analyses, experience, and the observation of behavior over time.
>
> *Note 2:* The security perspective places focus on credible evidence used to obtain assurance, substantiate trustworthiness, and assess risk.

facility
Physical means or equipment for facilitating the performance of an action (e.g., buildings, instruments, tools). [4]

flaw
Imperfection or defect.

generalized reference monitor concept
An abstract model of the necessary and sufficient properties that must be achieved by any mechanism that enforces a constraint. [36] [37]

incident
Anomalous or unexpected event, set of events, condition, or situation at any time during the life cycle of a project, product, service, or system. [4]

information
Knowledge that is exchangeable amongst users, about things, facts, concepts, and so on, in a universe of discourse. [80]

> *Note:* Although information will necessarily have a representation form to make it communicable, it is the interpretation of this representation (the meaning) that is relevant. The representation form is arguably considered data.

information item
Separately identifiable body of information that is produced, stored, and delivered for human use. [81]

information system
A discrete set of information resources organized for the collection, processing, maintenance, use, sharing, dissemination, or disposition of information. [82]

See *system*.

interface
Wherever two or more logical, physical, or both system elements or software system elements meet and act on or communicate with each other. [4]

interoperating system
System that exchanges information with the system of interest and uses the information that has been exchanged. [4]

integrity
Quality of being complete and unaltered. [83]

life cycle
Evolution of a system, product, service, project, or other human-made entity from conception through retirement. [4]

life cycle model
Framework of processes and activities concerned with the life cycle that may be organized into stages, which also acts as a common reference for communication and understanding. [4]

life cycle security concepts
The processes, methods, and procedures associated with the system throughout its life cycle and provide distinct contexts for the interpretation of system security. Life cycle security concepts apply during program management, development, engineering, acquisition, manufacturing, fabrication, production, operations, sustainment, training, and retirement.

likelihood
Chance of something happening. [28]

margin
A spare amount or measure or degree allowed or given for contingencies or special situations. The allowances carried to account for uncertainties and risks. [39]

See *design margin* and *operational margin*.

mechanism
A process or system that is used to produce a particular result.

The fundamental processes involved in or responsible for an action, reaction, or other natural phenomenon.

A natural or established process by which something takes place or is brought about.

> *Note 1:* Generally, a means to an end.
>
> *Note 2:* A mechanism can be technology- or non-technology-based (e.g., apparatus, device, instrument, procedure, process, system, operation, method, technique, means, or medium).

See *security mechanism*.

module
Program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading.

Discrete and identifiable element with a well-defined interface and well-defined purpose or role whose effect is described as relations among inputs, outputs, and retained state. [24]

monitoring
Continual checking, supervising, critically observing or determining the status in order to identify change from the performance level required or expected. [28]

operational concept
Verbal and graphic statement of an organization's assumptions or intent in regard to an operation or series of operations of a specific system or a related set of specific new, existing, or modified systems. [77]

> *Note:* The operational concept is designed to give an overall picture of the operations using one or more specific systems or set of related systems in the organization's operational environment from the perspectives of users and operators.

See *concept of operations*.

operational environment
Context determining the setting and circumstance of all influences on a delivered system.

> *Note:* Operational environments include physical (e.g., land, air, maritime, space) and cyberspace contexts.

operational margin
The margin that is designed explicitly to provide space between the worst normal operating condition and the point at which failure occurs (derives from physical design margin). [3] [6]

operator
Individual or organization that performs the operations of a system. [4]

> *Note 1:* The role of operator and the role of user can be vested, simultaneously or sequentially, in the same individual or organization.
>
> *Note 2:* An individual operator combined with knowledge, skills, and procedures can be considered an element of the system.
>
> *Note 3:* An operator may perform operations on a system that is operated or of a system that is operated, depending on whether or not operating instructions are placed within the system boundary.

organization
Group of people and facilities with an arrangement of responsibilities, authorities, and relationships. [4] [78]

> *Note:* An identified part of an organization (even as small as a single individual) or an identified group of organizations can be regarded as an organization if it has responsibilities, authorities, and relationships. A body of persons organized for some specific purpose – such as a club, union, corporation, or society – is an organization.

outcome
Result of the performance (or non-performance) of a function or process(es). [84]

party
Organization entering into an agreement. [4]

penetration testing
Testing used in vulnerability analysis for vulnerability assessment, trying to reveal vulnerabilities of the system based on the information about the system gathered during the relevant evaluation activities. [85]

problem
Difficulty, uncertainty, or otherwise realized and undesirable event, set of events, condition, or situation that requires investigation and corrective action. [4]

process
Set of interrelated or interacting activities that use inputs to deliver an intended result. [78]

process purpose
High-level objective of performing the process and the likely outcomes of effective implementation of the process. [4]

Note: The purpose of implementing the process is to provide benefits to the stakeholders.

process outcome
Observable result of the successful achievement of the process purpose. [86]

product
Result of a process. [78]

Note: There are four generic product categories: hardware (e.g., engine mechanical part); software (e.g., computer program); services (e.g., transport); and processed materials (e.g., lubricant). Hardware and processed materials are generally tangible products, while software or services are generally intangible.

project
Endeavor with defined start and finish criteria undertaken to create a product or service in accordance with specified resources and requirements. [4]

Note: A project is sometimes viewed as a unique process comprising co-coordinated and controlled activities and composed of activities from the Technical Management and Technical Processes defined in this document.

protection needs
Informal statement or expression of the stakeholder security requirements focused on protecting information, systems, and services associated with mission and business functions throughout the system life cycle.

Note: Requirements elicitation and security analyses transform the protection needs into a formalized statement of stakeholder security requirements that are managed as part of the validated stakeholder requirements baseline.

qualification
Process of demonstrating whether an entity is capable of fulfilling specified requirements. [86]

quality assurance
Part of quality management focused on providing confidence that quality requirements will be fulfilled. [78]

quality characteristic
Inherent characteristic of a product, process, or system related to a requirement. [78]

Note: Critical quality characteristics commonly include those related to health, safety, security, assurance, reliability, availability, and supportability.

quality management
Coordinated activities to direct and control an organization with regard to quality. [78]

reference monitor concept
An abstract model of the necessary and sufficient properties that must be achieved by any mechanism that performs an access mediation control function. [21] [37]

reference validation mechanism
An implementation of the reference monitor concept that validates each access to resources against a list of authorized accesses allowed. [37]

requirement
Statement that translates or expresses a need and its associated constraints and conditions. [31]

A condition or capability that must be met or possessed by a system or system element to satisfy an agreement, standard, specification, or other formally imposed documents. [131]

requirements engineering
An interdisciplinary function that mediates between the domains of the acquirer and supplier to establish and maintain the requirements to be met by the system, software, or service of interest. [31]

> *Note:* Requirements engineering is concerned with discovering, eliciting, developing, analyzing, verifying, validating, managing, communicating, and documenting requirements.

resource
Asset used or consumed during the execution of a process. [4]

> *Note 1:* Includes diverse entities, such as funding, personnel, facilities, capital equipment, tools, and utilities, such as power, water, fuel, and communication infrastructures.
>
> *Note 2:* Resources include those that are reusable, renewable, or consumable.

retirement
Withdrawal of active support by the operation and maintenance organization, partial or total replacement by a new system, or installation of an upgraded system. [4]

risk
Effect of uncertainty on objectives. [28]

> *Note 1:* An effect is a deviation from the expected, positive or negative. A positive effect is also known as an opportunity.
>
> *Note 2:* Objectives can have different aspects (i.e., financial, health and safety, and environmental goals) and can apply at different levels (i.e., strategic, organization-wide, project, product, and process).
>
> *Note 3:* Risk is often characterized by reference to potential events and consequences or a combination of these.
>
> *Note 4:* Risk is often expressed in terms of a combination of the consequences of an event (including changes in circumstances) and the associated likelihood of occurrence.
>
> *Note 5:* Uncertainty is the state, even partial, of deficiency of information related to the understanding or knowledge of an event, its consequence, or likelihood.

risk analysis
Process to comprehend the nature of risk and to determine the level of risk. [28]

risk assessment
Overall process of risk identification, risk analysis, and risk evaluation. [28]

risk criteria
Terms of reference against which the significance of a risk is evaluated. [28]

risk evaluation
Process of comparing the results of risk analysis with risk criteria to determine whether the risk and/or its magnitude is/are acceptable or tolerable. [28]

risk identification
Process of finding, recognizing, and describing risks. [28]

risk management
Coordinated activities to direct and control an organization with regard to risk. [28]

risk tolerance
The organization or stakeholder's readiness to bear the risk after risk treatment in order to achieve its objectives. [28]

Note: Risk tolerance can be influenced by legal or regulatory requirements.

risk treatment
Process to modify risk. [28]

safety
Expectation that a system does not, under defined conditions, lead to a state in which human life, health, property, or the environment is endangered. [86]

security
Freedom from those conditions that can cause the loss of assets with unacceptable consequences.

security architecture
A set of physical and logical security-relevant representations (i.e., views) of system architecture that conveys information about how the system is partitioned into security domains and makes use of security-relevant elements to enforce security policies within and between security domains based on how data and information must be protected.

Note: The security architecture reflects security domains, the placement of security-relevant elements within the security domains, the interconnections and trust relationships between the security-relevant elements, and the behavior and interactions between the security-relevant elements. The security architecture, like the system architecture, may be expressed at various levels of abstraction and with different scopes.

security design order of precedence
A design approach for minimizing the design basis for loss potential and using architectural features to provide structure for implementing engineered security features and devices.

security domain
Set of assets and resources subject to a common security policy. [85]

Note: A security domain is defined by rules (policy) for users, processes, systems, and services that apply to activities within the domain and activities with similar entities in other domains.

security function
The capability provided by the system or a system element. The capability may be expressed generally as a concept or specified precisely in requirements.

security mechanism
A device or method for achieving a security-relevant purpose.

security policy
A set of rules that governs all aspects of security-relevant system and system element behavior.

Note 1: System elements include technology, machine, and human elements.

Note 2: Rules can be stated at high levels of abstraction (e.g., an organizational policy that defines the acceptable behavior of employees in performing their mission and business functions) or at low levels of abstraction (e.g., an operating system policy that defines the acceptable behavior of executing processes and the use of resources by those processes).

security relevance
The functions or constraints that are relied upon to directly or indirectly meet protection needs.

Note: The term *security relevance* has been used to differentiate the role of system functions that singularly or in combination exhibit behavior, produce an outcome, or provide a capability to enforce authorized and intended system behavior or outcomes.

security requirement
A requirement that has security relevance.

security risk
The effect of uncertainty on objectives pertaining to asset loss and the associated consequences. [28]

Note: [28] defines risk as the effect of uncertainty on objectives. Furthermore, risk can be either positive or negative.

security service
A security capability or function provided by an entity.

security specification
The requirements for the security-relevant portion of the system.

Note: The security specification may be provided as a separate document or may be captured with a broader specification.

self-protection
The protection provided by an entity to ensure its own correct behavior and function despite adversity.

Note: While an entity would ideally be able to provide all the self-protection necessary, in practice entities are limited in the extent that they can provide for their own protection without depending on one or more other entities.

service
Performance of activities, work, or duties. [4]

Note 1: A service is self-contained, coherent, discrete, and can be composed of other services.

Note 2: A service is generally an intangible product.

situational awareness
Perception of elements in the system and/or environment and a comprehension of their meaning, which could include a projection of the future status of perceived elements and the uncertainty associated with that status. [87]

specification
An information item that identifies, in a complete, precise, verifiable manner, the requirements, design, behavior, or other expected characteristics of a system, service, or process. [128]

See *security specification*.

stage
Period within the life cycle of an entity that relates to the state of its description or realization. [4]

Note 1: As used in this document, stages relate to major progress and achievement milestones of the entity throughout its life cycle.

Note 2: Stages often overlap.

stakeholder
Individual or organization having a right, share, claim, or interest in a system or in its possession of characteristics that meet their needs and expectations. [4]

stakeholder (system)
Individual, team, organization, or classes thereof having an interest in a system. [71]

strength of function
Criterion expressing the minimum efforts assumed necessary to defeat the specified security behavior of an implemented security function by directly attacking its underlying security mechanisms.

Note 1: Strength of function has a prerequisite that assumes that the underlying security mechanisms are correctly implemented. The concept of strength of function may be equally applied to services or other capability-based abstraction provided by security mechanisms.

Note 2: The term *robustness* combines the concepts of assurance of correct implementation with strength of function to provide finer granularity in determining the trustworthiness of a system.

susceptibility
The inability to avoid adversity.

supplier
Organization or an individual that enters into an agreement with the acquirer for the supply of a product or service. [4]

Note 1: Other terms commonly used for supplier are contractor, producer, seller, or vendor.

Note 2: The acquirer and the supplier are sometimes part of the same organization.

system
An arrangement of parts or elements that together exhibit behavior or meaning that the individual constituents do not. Systems can be physical or conceptual, or a combination of both. [3] [4]

Note 1: A system is sometimes considered as a product or as the services that it provides.

Note 2: In practice, the interpretation of its meaning is frequently clarified by the use of an associative noun (e.g., aircraft system). Alternatively, the word "system" is substituted simply by a context-dependent synonym (e.g., aircraft), though this potentially obscures a system principles perspective.

Note 3: A complete system includes all associated equipment, facilities, material, computer programs, services, firmware, technical documentation, and personnel required for operations and support to the degree necessary for self-sufficient use in its intended environment.

system element
Member of a set of elements that constitute a system. [4]

Note: A system element is a discrete part of a system that can be implemented to fulfill specified requirements.

system of interest
System whose life cycle is under consideration. [4]

system of systems
System of interest whose system elements are themselves systems; typically, these entail large-scale interdisciplinary problems with multiple, heterogeneous, distributed systems. [15]

Set of systems or system elements that interact to provide a unique capability that none of the constituent systems can accomplish on its own. [88]

system context
The specific system elements, boundaries, interconnections, interactions, and operational environment that define a system.

system life cycle
Period that begins when a system is conceived and ends when the system is no longer available for use. [24]

See *life cycle stages*.

system security requirement
System requirement that has security relevance. System security requirements define the protection capabilities provided by the system, the performance and behavioral characteristics exhibited by the system, and the evidence used to determine that the system security requirements have been satisfied.

> *Note 1:* Due to the complexity of system security, system security requirements have several types and purposes, including (1) structural security requirements that express the passive aspects of the protection capability provided by the system architecture and (2) functional security requirements that express the active aspects of the protection capability provided by the engineered features and devices (e.g., security mechanisms, inhibits, controls, safeguards, overrides, and countermeasures).
>
> *Note 2:* Each system security requirement is expressed in a manner that makes verification possible via analysis, observation, test, inspection, measurement, or other defined and achievable means.

systems engineering
A transdisciplinary and integrative approach to enable the successful realization, use, and retirement of engineered systems, using systems principles and concepts, and scientific, technological, and management methods. [3]

Interdisciplinary approach governing the total technical and managerial effort required to transform a set of stakeholder needs, expectations, and constraints into a solution and to support that solution throughout its life. [24]

systems security engineer
Individual who practices the discipline of systems security engineering, regardless of their formal title. Additionally, the term *systems security engineer* refers to multiple individuals who operate on the same team or cooperating teams.

systems security engineering
A transdisciplinary and integrative approach to enable the successful secure realization, use, and retirement of engineered systems using systems, security, and other principles and concepts, as well as scientific, technological, and management methods. Systems security engineering is a subdiscipline of systems engineering.

tampering
An intentional but unauthorized act resulting in the modification of a system, components of systems, its intended behavior, or data. [89]

task
Required, recommended, or permissible action intended to contribute to the achievement of one or more outcomes of a process. [4]

threat
Potential cause of unacceptable asset loss and the undesirable consequences or impact of such a loss.

> *Note:* The specific causes of asset loss can arise from a variety of conditions and events related to adversity, typically referred to as disruptions, hazards, or threats. Regardless of the specific term used, the basis of asset loss constitutes all forms of intentional, unintentional, accidental, incidental, misuse, abuse, error, weakness, defect, fault, and/or failure events and associated conditions.

traceability
Discernible association among two or more logical entities, such as requirements, system elements, verifications, or tasks. [90]

traceability matrix
A matrix that records the relationship between two or more products of the development process (e.g., a matrix that records the relationship between the requirements and the design of a given software component). [24]

trade-off
Decision-making actions that select from various requirements and alternative solutions on the basis of net benefit to the stakeholders. [4]

trade-off analysis
Determining the effect of decreasing one or more key factors and simultaneously increasing one or more other key factors in a decision, design, or project.

transdisciplinary
Creating a unity of intellectual frameworks beyond the disciplinary perspectives. [135]

trust
A belief that an entity meets certain expectations and therefore, can be relied upon. [39]

Note: The term *belief* implies that trust may be granted to an entity whether the entity is trustworthy or not.

trust relationship
An agreed upon relationship between two or more system elements that is governed by criteria for secure interaction, behavior, and outcomes relative to the protection of assets.

Note: This refers to trust relationships between system elements implemented by hardware, firmware, and software.

trustworthiness
Worthy of being trusted to fulfill whatever critical requirements may be needed for a particular component, subsystem, system, network, application, mission, enterprise, or other entity. [2]

Note: From a security perspective, a trustworthy system meets specific security requirements in addition to meeting other critical requirements.

trustworthy
The degree to which the behavior of a component is demonstrably compliant with its stated requirements.

user
Individual or group that interacts with a system or benefits from a system during its utilization. [91]

Note: The role of user and the role of operator are sometimes vested, simultaneously or sequentially, in the same individual or organization.

validation
Confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled. [78]

Note: A system is able to accomplish its intended use, goals, and objectives (i.e., meet stakeholder requirements) in the intended operational environment. The right system was built.

verification
Confirmation, through the provision of objective evidence, that specified requirements have been fulfilled. [78]

Note: Verification is a set of activities that compares a system or system element with the required characteristics. This includes but is not limited to specified requirements, design description, and the system itself. The system was built correctly.

view
Representation of a whole system from the perspective of a related set of concerns. [92]

Note: A view can cover the entire system being examined or only a part of that system.

viewpoint
Specification of the conventions for constructing and using a view. [92]

vulnerability
A weakness that can be exploited or triggered to produce an adverse effect.

The inability to withstand adversity.

> *Note:* Vulnerability can exist in anywhere throughout the life cycle of a system, such as in the CONOPS, procedures, processes, requirements, design, implementation, utilization, and sustainment of the system.

weakness
Defect or characteristic that may lead to undesirable behavior. [93]

> *Note:* Examples include a missing requirement or specification; architectural or design flaw; implementation weakness, including hardware or software defect; or the use of an outdated or deprecated function, including outdated cryptographic algorithms.

Appendix C. Security Policy and Requirements

This appendix discusses security policy and requirements considerations[57] in support of Appendix D, Appendix E, and Appendix H. Covered topics include the rules and scope of control for security policy (Section C.1), stakeholder and system security requirements (Section C.2), and the relationship among security requirements, policy, and mechanisms (Section C.3).

C.1. Security Policy

A *security policy* is a set of rules (Section C.1.1) that governs behavior and outcomes within a defined scope of control (Section C.1.2). The policy generally includes a set of policies that reflect the needs and expectations established by an authority with a specific scope and purpose (Section C.1.2). The policy rules have a hierarchy, from security policy top-level objectives that are refined and allocated to organizational security policies, which in turn are refined and allocated to system security policies.

C.1.1. Rules

Security policy rules are stated in terms of authorized relationships that involve subjects (i.e., active entities) and objects (i.e., passive entities). The rules govern the operations that a subject can perform or invoke on other subjects (i.e., subject-to-subject operations) and the operations that a subject can perform or invoke on objects (i.e., subject-to-object operations). The rules must be accurate, consistent, compatible, and complete with respect to stakeholder security objectives within the defined scope of control. Inaccurate, inconsistent, incompatible, or incomplete rule sets will allow undesired behavior and outcomes.

C.1.2. Scope of Control

Security policies reflect and are derived from laws, directives, regulations, life cycle concepts,[58] requirements, or stakeholder objectives. Each policy includes a scope of control that establishes the bounds within which the policy applies. A typical scope of applicability includes:

- **Security Policy (Protection) Objectives:** A set of objectives that captures a preferred state or what is to be achieved. These objectives include assets to be protected, statements of intent to protect the assets within the specific scope of stakeholder responsibility, and the protection scope. Security policy objectives are the basis for deriving all other security policy forms.
- **Organizational Security Policy:** A set of rules[59] that regulates how an organization achieves its objectives. The rules provide individuals with a reasonable ability to determine whether their actions either violate or comply with the security policy. Organizational security policy defines the individual's behavior in performing their missions and business functions and is used for developing processes and procedures.

[57] This appendix discusses policy in a manner that suggests policy precedes engineering. However, policy may need to be modified to align with the capabilities of the delivered as-is system.

[58] Life cycle concepts include operation, sustainment, evolution, maintenance, training, startup, and shutdown.

[59] The rules may be captured in laws and practices.

- **System Security Policy:** A policy that specifies the system security capability. It is the set of restrictions and properties that specifies how a system enforces or contributes to enforcing organizational security policy.
- **Personnel Security Policy:** A policy that defines the expectations of personnel.[60] These include the behaviors of the personnel using or sustaining the system.

Security policy goes through an iterative refinement process that decomposes an abstract statement of security policy into more specific statements of security policy. The refinement occurs in parallel with requirements allocation and decomposition. Figure 11 illustrates security policy allocation across the organization.

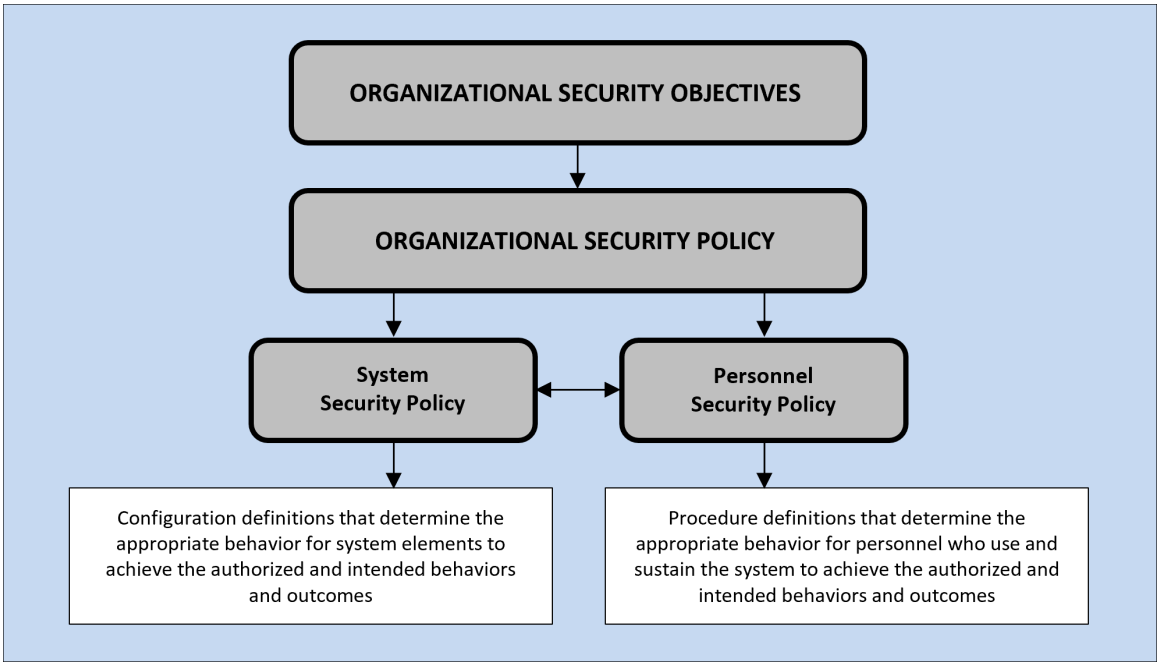

Fig. 11. Allocation of Security Policy Responsibilities

C.2. Security Requirements

A requirement is a statement that translates or expresses a specific need and its associated constraints and conditions.[61] Security requirements translate or express protection needs (Section 3.7), associated constraints, and associated conditions. The constraints also reflect concerns about the system functions, system architecture, and design to ensure that they are specified in a manner that avoids and reduces susceptibilities, defects, flaws, and weaknesses (Section 3.8) and is consistent with the needs of active security functions.

Requirements can be categorized as (1) *stakeholder requirements* that address the need to be satisfied in a design-independent manner and (2) *system requirements* that express the specific

[60] These expectations often cover personnel actions that may expose them to negative external influences (e.g., social media use).
[61] General requirements and definition processes are described in sources such as [31]and [32].

solution that will be delivered (design-dependent manner). Figure 12 illustrates the two types of requirements and their relationship to the verification and validation of the system.

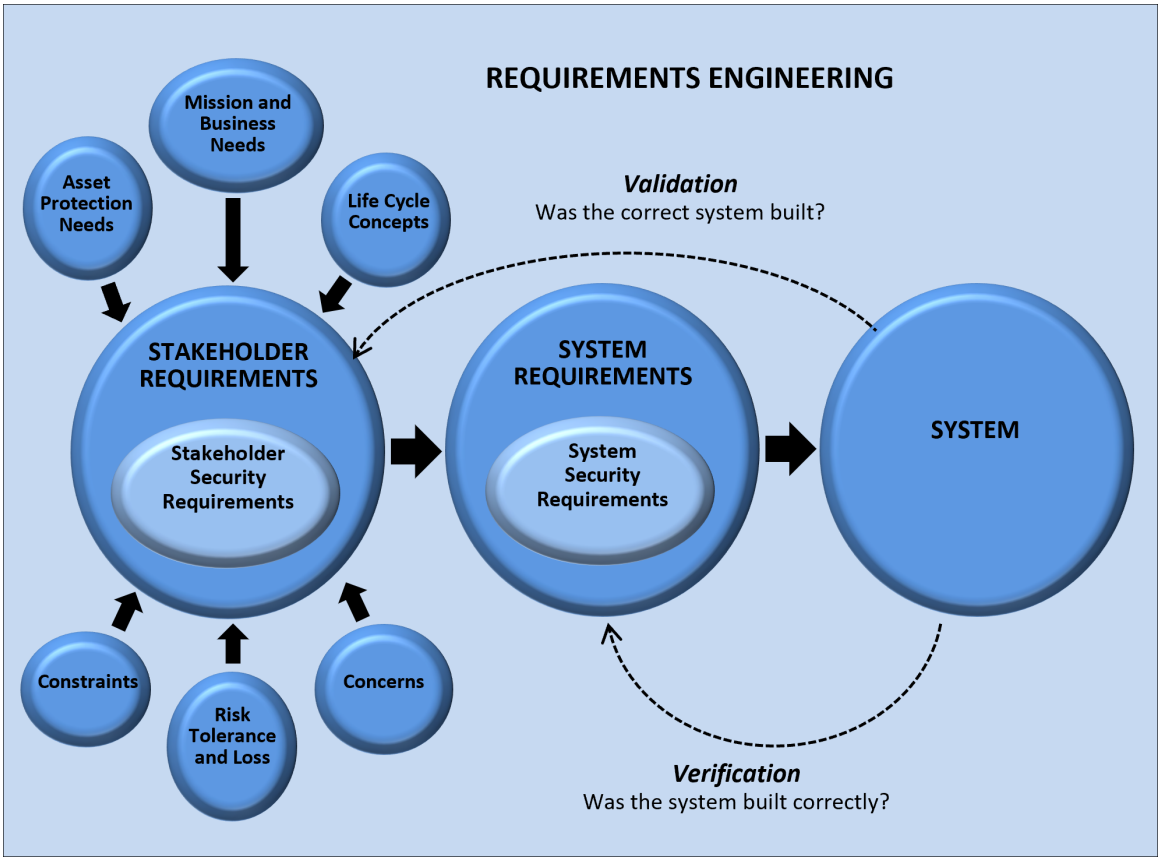

Fig. 12. Stakeholder and System Requirements

Security requirements and security-relevant constraints and conditions on other requirements are informed by various items, such as those pictured in Figure 13.

C.2.1. Stakeholder Security Requirements

Stakeholder security requirements are those stakeholder requirements that have security relevance. These requirements specify:

- The protection needed for the mission or business, data, information, processes, functions, humans, and system assets
- The roles, responsibilities, and security-relevant actions of individuals who perform and support the mission or business processes
- The interactions between the security-relevant solution elements
- The assurance that is to be obtained in the security solution

Systems security considerations within activities and tasks such as those described in Appendices H, I, J, and K provide the security perspective to ensure that stakeholder security requirements are included in the stakeholder requirements and that the stakeholder security requirements are consistent with all other stakeholder requirements.

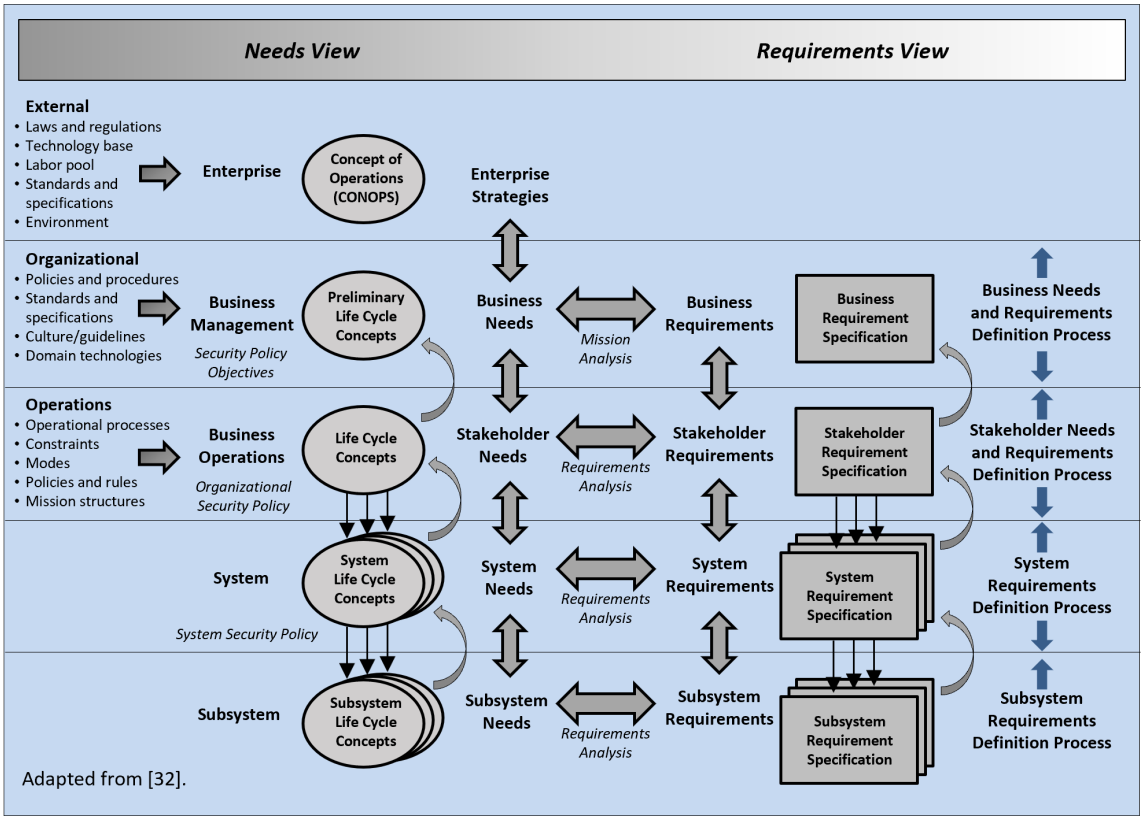

Fig. 13. Entities that Affect Security Requirements Development

C.2.2. System Security Requirements

System requirements specify the technical view of a system or solution that meets the specified stakeholder needs. The system requirements are a transformation of the validated stakeholder requirements. System requirements specify what the system or solution must do to satisfy the stakeholder requirements. System security requirements are those system requirements that have security relevance. These requirements define:

- The protection capabilities provided by the security solution
- The performance and behavioral characteristics exhibited by the security solution
- Assurance processes, procedures, and techniques
- Constraints on the system and the processes, methods, and tools used to realize the system

- The evidence required to determine that the system security requirements have been satisfied[62]

Due to the complexity of system security, system security requirements have several types and purposes, including (1) structural security requirements that express the passive aspects of the protection capability provided primarily by the system architecture and (2) functional security requirements that express the active aspects of the protection capability provided by engineered features and devices (e.g., security mechanisms, controls, safeguards, inhibits, overrides, and countermeasures). Decomposition of the system security requirements is accomplished as part of the system requirements decomposition and is consistent with the different levels of hierarchical abstraction and forms of the system requirements.

SYSTEM STATES, POLICY, AND REQUIREMENTS

Systems operate in secure, insecure, and indeterminant states (Section 3.2). System security policy and system requirements account for these states and the state transitions, including those that reflect the design principles of *Protective Failure* and *Protective Recovery*. For example, requirements capture needs to (1) detect insecure system states; (2) detect a transition that will result in an insecure state; (3) transition to a secure halt state; (4) recover to a reconstituted, reconfigured, or alternative secure operational mode; and if necessary, (5) continue operating within insecure or indeterminant states when other needs override protection needs.

C.3. Distinguishing Requirements, Policy, and Mechanisms

The terms *requirements*, *policy*, and *mechanisms* are often used in an abstract manner that allows them to be considered as synonyms. However, when these terms are used in the context of engineering trustworthy secure systems, they are distinct in their meaning and importance to specifying, realizing, utilizing, and sustaining systems.

The security policy states the behavior that is necessary to achieve a secure condition, whereas a security mechanism is a means to achieve the necessary behavior. The distinction between security policy and security mechanism extends to differentiating security requirements from security policy. Security requirements specify the capability, behavior, and quality attributes exhibited and possessed by security mechanisms as well as the constraints on each. Security policy specifies how the security mechanisms must behave in an operational context and the constraints on those behaviors. From the system standpoint, a human is a system element and may serve as a security mechanism. Therefore, the human is expected to behave as stated by relevant security policy and security requirements.

Requirements, policies, and mechanisms have an important dependency relationship. System security requirements specify the capabilities and behaviors that a security mechanism can

[62] Each system security requirement, like any system requirement, is expressed in a manner that makes verification possible via inspection, analysis, demonstration, testing, or other defined and achievable means [31].

provide. A security policy specifies the aspects that a mechanism must enforce to achieve organizational objectives. This means that a secure system cannot be achieved if the security requirements do not fully specify the minimal capability necessary to enforce the security policy. It also means that the satisfaction of requirements alone does not result in a secure system. Verification and validation activities must be done separately and coordinated to ensure the individual and combined correctness and effectiveness of the requirements and policy.

Figure 14 illustrates the significance of the consistency relationship that must be maintained across interacting security requirements, security policy, and security mechanisms.

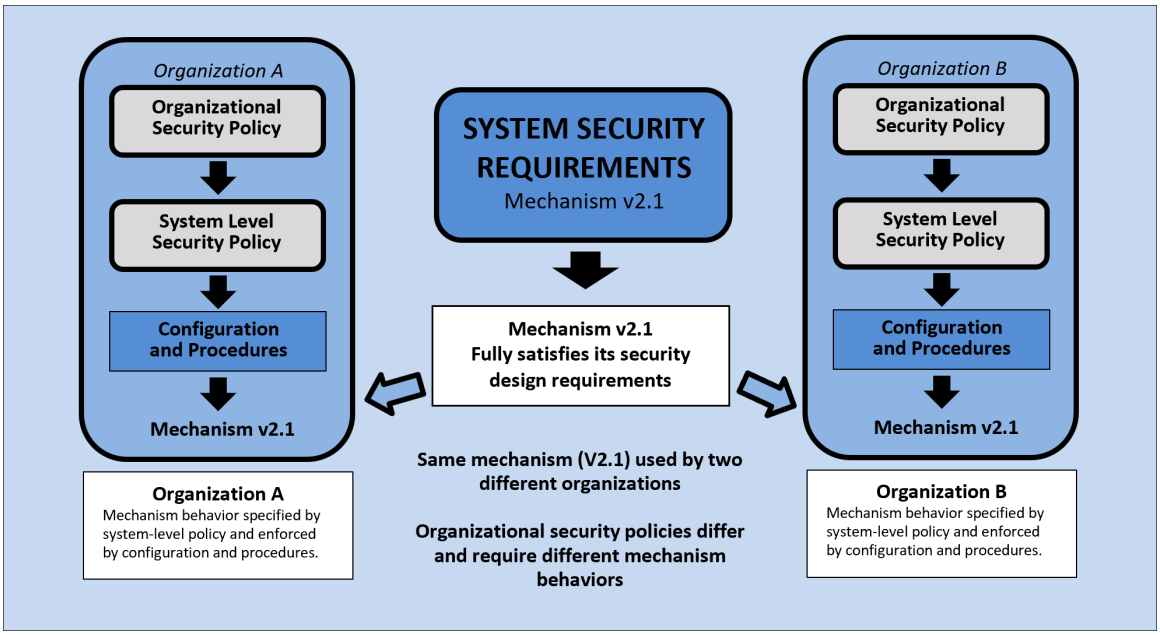

Fig. 14. Relationship between Mechanisms and Security Policy Enforcement

Note that a security mechanism that fully satisfies its system security requirements may be deemed capable of enforcing the security policy that is defined for two different organizations. Each organization will use the same mechanism and configure it to behave in a manner that enforces the rules of their organizational security policy. However, if the organizations were to switch mechanisms and keep the same configuration of the mechanism, they would achieve uncertain results (unless their security policy objectives required the exact same configuration of the mechanism). From this, the following conclusions may be drawn:

- Requirements express both the security protections to be provided by security mechanisms and the security-informed constraints to be enforced by security mechanisms.
- Security policy determines the behavior and outcomes that are deemed *secure*.
- For a mechanism to be deemed secure, the mechanism's capability requirements must be consistent with security policy enforcement rules; the mechanism must satisfy the security requirements; and the mechanism must be configured to behave in a manner defined by the organizational security policy.

Appendix D. Trustworthy Secure Design

This appendix discusses the approach and considerations for applying technical[63] concepts of a trustworthy secure system design. The concepts described provide a balanced and integrated approach that optimally protects against asset loss. The discussions include the design approach for trustworthy systems (Section D.1), authorized and intended behaviors and outcomes of the system (Section D.2), security design order of precedence (Section D.3), and functional design and trade space considerations (Section D.4).

A principled system design strengthens trustworthiness claims [2]. The concepts in this appendix and the principles in Appendix E provide a sound basis for reasoning about a system and enable the demonstration of system trustworthiness. Applying concepts and principles and using other enablers (e.g., standards, specifications, design patterns, security policy models, cryptographic algorithms, security protocols, strength of mechanism, and known adversities) should be planned for, appropriately scoped, and revisited throughout the system life cycle and engineering effort.

Finally, trustworthiness and assurance needs and considerations also inform trustworthy secure design. Appendix F provides further discussion of the concepts of trustworthiness and assurance.

TRUSTWORTHY SECURE DESIGN

Trustworthy secure design is a means to provide stakeholders with the confidence that their conflicting capability needs, concerns, priorities, and constraints are satisfied.

D.1. Design Approach for Trustworthy Systems

The design approach for engineering trustworthy secure systems is intended to establish and maintain the ability to deliver system capabilities at an acceptable level of performance[64] while minimizing the occurrence and extent of loss. This approach provides a system structure for optimal employment of the engineered features and devices. The system design must provide the intended behaviors and outcomes, avoid the unintended behaviors and outcomes, prevent loss, and limit loss when it occurs. A trustworthy secure design includes a situational awareness capability and margin[65] to account for adversity due to the unknowns and uncertainty inherent in the system and in its operational environment. The situational awareness capability helps to determine accountability for the actions of all users and entities (e.g., audit, logging, event

[63] Note that human factor elements of trust are not discussed. A system may be trustworthy, but a user may not trust it. Similarly, a user may trust an untrustworthy system.

[64] An acceptable level of performance lies between the minimum threshold of acceptability and the objective of maximum performance. This level may vary across operational or system states and modes (e.g., patrolling in clear weather versus severe weather conditions), may vary across contingency conditions (e.g., normal, degraded), and may be subject to operational priorities (e.g., search and rescue, manhunt).

[65] The term *margin* refers to a spare amount, measure, or degree allowed or given for contingencies or special situations. The allowances are carried to account for uncertainties and risks. Two types of margins are used in systems engineering: design margin and operational margin. See the design principle of *Loss Margins*.

recording) and detect pending and actual failure (e.g., by crossing the threshold of the margins that have been established). The design principle of *Anomaly Detection* embodies this capability.

The design approach includes the following elements:[66]

- Define the intended behaviors and outcomes for the system[67]
- Identify the system states and conditions that reflect the intended behaviors and outcomes
- Identify the system states and conditions that potentially lead to loss in the system
- Select and alter the system design to prevent loss to the extent practicable (preferred) and limit the loss that does occur (where, when, and to the extent necessary and practicable)
- Iterate the above elements to address how the functions that serve to prevent or limit loss may fail due to intentional or unintentional reasons

Figure 15 illustrates the steps in the design approach in the context of the Systems Security Engineering Framework described in Chapter Four.

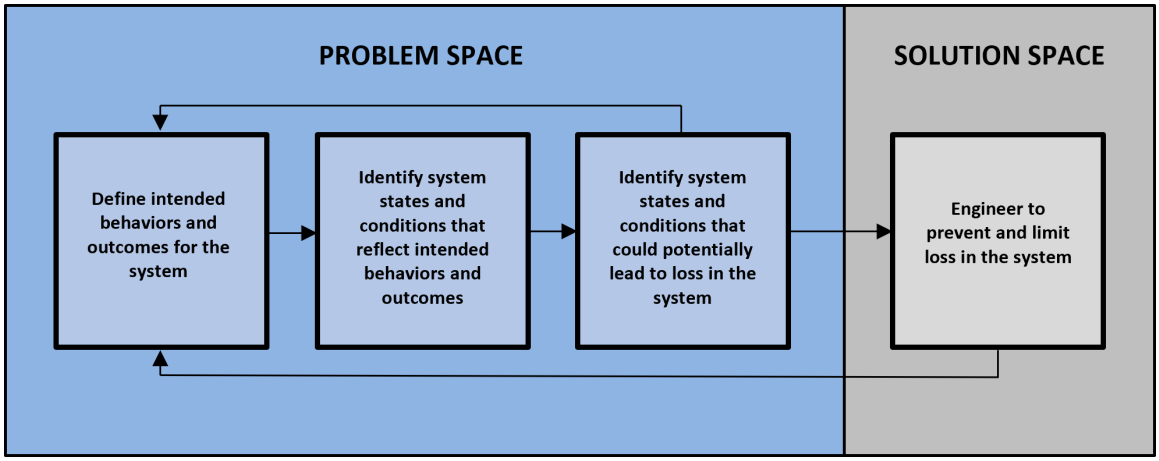

Fig. 15. Design Approach in a Systems Security Engineering Framework

The approach to trustworthy design includes both preemptive and reactive aspects. These mutually reinforcing aspects provide the protection needed to achieve only the authorized and intended behaviors and outcomes. The preemptive aspect results in system features and system actions taken to prevent and limit loss before the loss occurs, while the reactive aspect results in system actions to limit loss and its effects once a loss has occurred. Figure 16 illustrates a balanced design strategy that includes preemptive and reactive aspects.

[66] These steps are useful in applying a system control concept for any loss-relevant emergent property (e.g., safety, resilience).

[67] This flow iterates through systems engineering as the system is decomposed. Subsequent iterations would apply within the elements that comprise the system of interest (i.e., the subsystems, assemblies, and components).

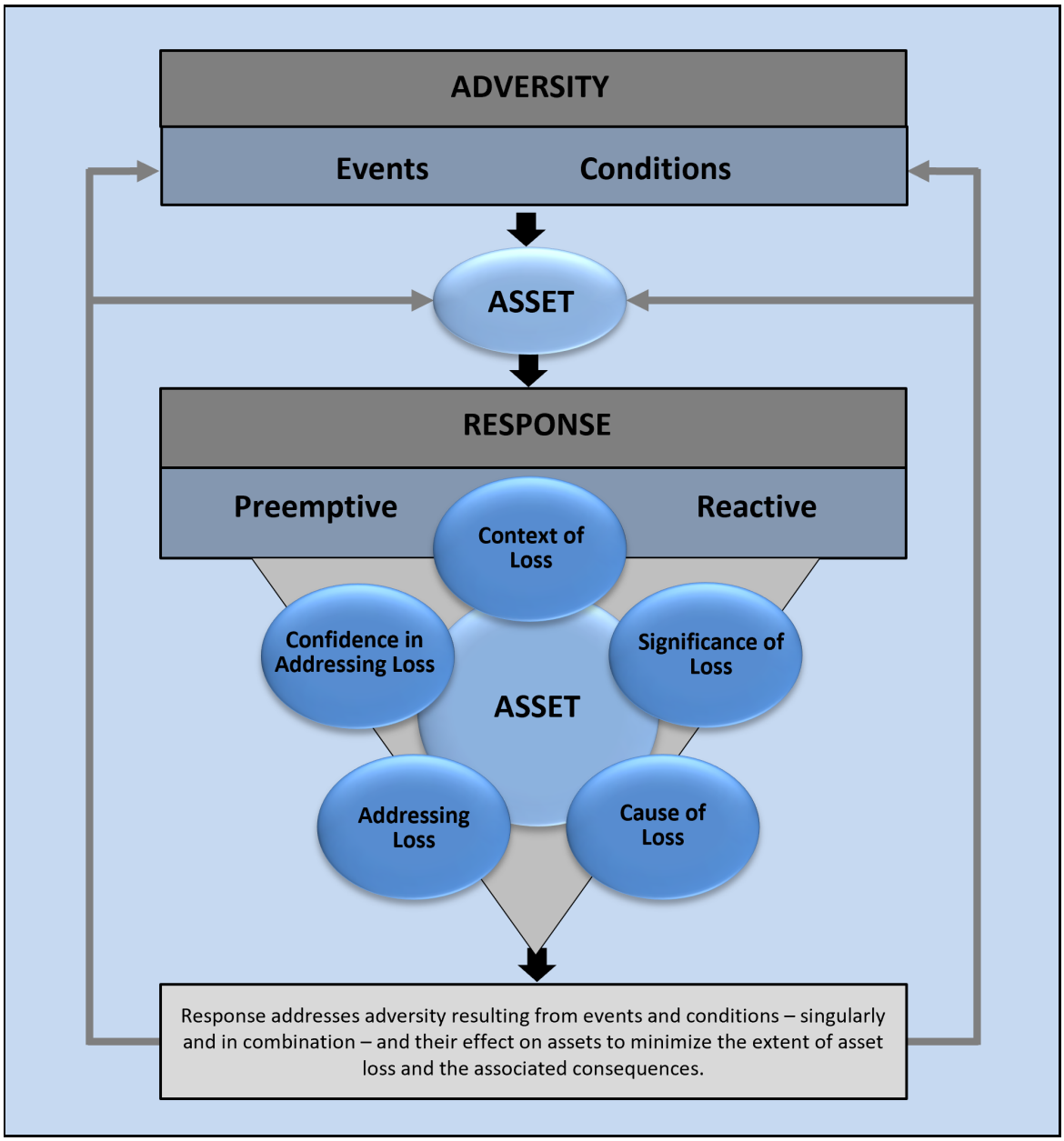

Fig. 16. Balanced Design Strategy for Achieving Trustworthy Secure Systems

The preemptive aspect recognizes the conditions under which loss may occur and addresses the scenarios before loss occurs (i.e., what can happen). If the loss does occur, the results are limited due to system features and actions taken in advance. The preemptive aspect is independent of any specific knowledge of attacks and attacker objectives, instead focusing on what is possible in the system's life cycle. The reactive aspect recognizes the limits of certainty about what can happen, and that new, unanticipated, and otherwise unforeseen adverse consequences will occur despite proactive planning and instituting of means and methods to control loss and the extent of its consequences. The reactive aspect promotes informed operational decision-making after the system is in use and a loss condition occurs, proactively providing an operations capability to address the loss condition and handle the loss. The reactive aspect complements the preemptive

aspect by providing an informed basis and means for an external entity (e.g., a human operator or system) to act when failures occur. Essentially, the reactive aspect is a proactive engineering activity about providing a reactive capability.

An effective design will optimize protection against loss to the extent practical while recognizing that losses will occur irrespective of the protections put in place. Optimization decisions across preemptive and reactive approaches must consider assets, stakeholders, concerns, and objectives. Achieving a proper mix requires establishing security objectives and conducting requirements elicitation and analysis to unambiguously ascertain the scope of security in terms of addressing failure and the associated consequences in its preemptive and reactive aspects.

D.2. Design Considering Emergence

A system is expected to deliver the required capabilities as authorized, as intended, and at the specified level of performance. It should not deliver unauthorized or unintended capabilities. One cause of unintended behaviors and outcomes lies with the concept of *emergence*. Emergence refers to the behaviors and outcomes that result from how individual system elements compose to form the system beyond the collection of behaviors and outcomes of the individual system elements. This composition is covered in the design principle of *Structured Decomposition and Composition* and illustrated in Figure 2.

Some emergence is desired and productive; other emergence is not desired or productive, creating unknown, unforeseen, or adverse effects.[68] Engineering trustworthy secure systems seeks to deliver only desired emergence. Trustworthiness judgments are based on the expectation that the system can satisfy the stated capability needs. To achieve this, the design must address emergence at all levels of system abstraction in terms of how the system is decomposed into its constituent elements and how those system elements compose to produce the system. This is covered in the design principle of *Compositional Trustworthiness*.

SECURITY IS EMERGENT

The objective of security is to achieve only authorized and intended system behaviors and outcomes. This requires a fundamental understanding of how individual system elements are composed into the system as a whole. Systems are designed from that basis of understanding to limit emergent behaviors and outcomes that are not specified (including desired and undesired unspecified behaviors and outcomes).

[68] Emergence may be described in terms of properties exhibited by entities only when attributed to the whole, not to any individual constituent element.

D.3. Security Design Order of Precedence

The security design order of precedence (SecDOP)[69] is a design approach with the objective of minimizing the system design basis for loss potential. SecDOP emphasizes the importance of establishing a secure structural context for the employment of engineered features and devices. Using a principled engineering approach, the SecDOP eliminates susceptibility, hazard, and vulnerability to the extent practicable, thereby eliminating the associated risk. For those cases in which susceptibility, hazard, or vulnerability cannot be eliminated, the SecDOP reduces the potential of experiencing a loss to an acceptable level within the constraints of cost, schedule, and performance. The SecDOP identifies design alternatives in order of decreasing effectiveness, thus enabling a maximized return on investment.

The SecDOP alternatives are:

- **Eliminate the potential for loss through design selection**

  Susceptibility, hazard, and vulnerability are eliminated by selecting a design or material alternative that completely removes susceptibility, hazard, and vulnerability and, thus, prevents loss.

  - *Example:* The design selected for a system function of interest results in the minimum number of external interfaces to other systems and the minimum number of internal interfaces that provide the required functions. Since every interface presents a potential for susceptibility, hazard, and vulnerability, a design selection with the minimum number of interfaces results in less susceptibility, hazard, and vulnerability than a design that includes additional and unnecessary internal and external interfaces.
  - *Note:* The design selection also considers the need to accommodate mechanisms that provide mediated access and trusted communication as these engineered features and devices are necessary for a secure system.

- **Reduce the potential for loss through design alteration**

  If adopting an alternative design or material to eliminate susceptibility, hazard, and vulnerability is not feasible, consider design changes or material selections that would reduce the frequency, potential, severity, and/or extent of loss caused by the susceptibility, hazard, or vulnerability.

  - *Example:* The selected design has interfaces and, therefore, has inherent susceptibilities, hazards, and vulnerabilities, both with respect to interactions with the environment and with respect to interactions within the system. The potential for loss can be reduced by altering design to segment functionality that interacts with external entities and influences from functionality strictly within system boundaries.
  - *Note:* The design alteration also considers the need to accommodate mechanisms that provide mediated access and trusted communication, as these engineered features and devices are necessary for a secure system.

[69] The security design order of precedence is inspired by the System Safety Design Order of Precedence, an optimized design approach for system safety described in [34].

- **Incorporate engineered features or devices to control the potential for loss**

  If preventing, limiting, or reducing the potential for loss through design alteration and material selection is not feasible or adequate, employ engineered features and devices to control loss associated with susceptibility, hazard, and vulnerability. In general, engineered features actively disrupt the loss scenario sequence and interactions, and devices reduce the potential, severity, and extent of loss.

  Two general types of engineered features and devices employed to address the potential for loss associated with the system function of interest are:

  - *Mandatory security features and devices:* Mandatory security features and devices apply foundational security principles to the interfaces. For example, each interface must have mediated access to control access to and use of the capability and data provided by the interface.
  - *Function-specific features and devices:* Function-specific security features and devices protect against a loss associated with the design's ability to meet functional requirements and performance parameters. Engineered features such as redundant data and control flows and redundant system elements can supplement the design selection to achieve the required protection. The system may also have engineered features that enable external entities to intervene in the system to address the potential, severity, or extent of loss.

- **Provide visibility and feedback to external entities**

  If design alteration, material selection, and engineered features and devices are not feasible or do not adequately lower the frequency, potential, severity, or extent of loss caused by the susceptibility, hazard, or vulnerability, employ engineered detection and feedback systems and warning devices to alert external entities to the presence of a susceptible, hazardous, or vulnerable condition; the occurrence of an event that will lead to a loss; or an actual loss event. External entities include operational personnel, monitoring systems, or other systems capable of responding.

  - *Example:* Anomaly detection features can be used to provide situational awareness data and warnings to system users.
  - *Note:* The visibility and feedback provided is not of value if the external entities are not able to respond appropriately. For example, the visibility and feedback must be provided in a timely manner and be in a form that can be interpreted correctly.

- **Incorporate signage, procedures, training, and proper equipment**

  Incorporate procedures, training, signage, and proper equipment where design alternatives, design changes, and engineered features and devices are not feasible and warning devices cannot adequately lessen the potential, severity, or extent of loss caused by the hazard, susceptibility, or vulnerability. Procedures and training include proper warnings and cautions and may prescribe the use of equipment. The use of signage, procedures, training, and equipment as the only means to reduce the potential, severity, or extent of loss should be avoided.

- *Example:* Procedures and training address the proper use of the system function of interest, the use of mediated access functions, and the relevant warnings, cautions, and warning systems.

ON USING SECURITY CONTROLS

Common practice in some communities is to select and overlay systems with security controls (i.e., management, operational, and technical safeguards or countermeasures) as a primary means to address security concerns [67] [68]. But as observed in [69], "Poor systems security engineering is very difficult to mitigate by overlaying security controls, whereas security controls overlaid on a sound, secure design can be quite effective."

The Security Design Order of Precedence, as part of a systems engineering practice, helps to ensure a proper integration of *technical* controls and *operational* controls.

D.4. Functional Design Considerations

This section describes the functional design considerations for trustworthy secure systems. These considerations include (1) assured functions that provide control enforcement, control decision, and control infrastructure; (2) design criteria for mechanisms; (3) security function failure analysis; (4) situational awareness; and (5) trade space considerations.

D.4.1. Roles for Security-Relevant Control

All functions have the potential to influence behaviors and outcomes beyond themselves and are relevant to security.[70] However, some functions have dedicated security purposes (e.g., functions that support audit capabilities). Examples include the protection control functions.

Protection control functions enforce or contribute to the control of or otherwise directly influence system or system element behaviors and outcomes. These functions may be characterized and evaluated by using the following designations:

- *Protection Control Decision Functions:* These functions make authorization decisions or take other actions for protection control enforcement functions. For example, a function that decides to grant or deny access to a resource based on a request (e.g., from a protection control enforcement function).
- *Protection Control Enforcement Functions:* These functions enforce a constraint to ensure that the system or system element exhibits only authorized and intended behaviors or

[70] Historically, the term *security relevance* has been used in secure system design and evaluation to differentiate the role of system functions that either singularly or in combination exhibit a behavior, produce an outcome, or provide a capability to enforce authorized and intended system behaviors or outcomes. However, from the security perspective (Section 3.8) and the possibility of loss due to weaknesses and defects in any system function, all functions have loss-related concerns and, thus, protection concerns.

outcomes. For example, a protection control enforcement function enforces a decision to grant or deny access to a resource.

- *Protection Control Infrastructure Functions:* These functions support and help protection control enforcement and control decision functions fulfill their purposes. The functions also provide data or services or perform operations upon which protection control enforcement and decision functions depend. For example, a protection control infrastructure function includes secure storage, secure communication, and anomaly detection mechanisms.

Other functions, including control functions for other purposes besides protection, can potentially adversely affect the correct operation of the protection control functions. For the purposes of secure design and evaluation, the functions are designated *other system functions*. Ideally, these functions should be non-interfering. This non-interference objective may be achieved through assurance with constraints on the requirements, architecture, design, and use of these functions.

System functions can be mapped to one or more protection control decision functions, protection control enforcement functions, or protection control infrastructure functions, or other system functions for the purpose of secure design and evaluation. The distinction guides and informs a principled design to limit interference among functions with confidence. Such confidence can be achieved by employing *Trustworthy System Control*, applying the design criteria described in Section D.4.2, and optimally placing a function in the system architecture to limit the side effects and interactions that may interfere with the protection control functions.

System analyses can determine the extent to which functions may interfere with other functions, including identifying any needed actions to increase assurance (Appendix F). For example, to satisfy a specific size or form-factor constraint, a system function mapped to a system function designated as “other” may occupy the same privilege domain as control enforcement, control decision, or control infrastructure functions, thereby elevating the privilege of that system function. If the size or form-factor constraint does not exist, it would be prudent to allocate that system function elsewhere to avoid giving the function elevated privilege. This would increase the assurance that the enforcement, decision, and infrastructure functions are isolated from the other parts of the system and would not be adversely impacted by their behavior or provide an avenue for attack.

D.4.2. Essential Design Criteria for Mechanisms

To effectively achieve the objectives of trustworthy secure design, mechanisms (i.e., engineered features and devices) must satisfy four essential design criteria. They must be non-bypassable, evaluatable, always invoked, and tamper-proof [35]. Generally, a design for any control function that provides protection should adhere to these criteria.[71] Table 3 briefly describes the essential design criteria.

[71] The argument that any control function should be non-bypassable, evaluatable, always invoked, and tamper-proof follows from an in-depth examination of Systems Theoretic Process Analysis (STPA) as described in [36], specifically the discussions on why controls may fail and how to address failure.

Table 3. Essential Design Criteria for Mechanisms

ESSENTIAL DESIGN CRITERIA	DESCRIPTION
NON-BYPASSABLE	The mechanism must not be circumventable.
EVALUATABLE	The mechanism must be sufficiently small and simple enough to be assessed to produce adequate confidence in the protection provided, the constraint (or control objective) enforced, and the correct implementation of the mechanism. The assessment includes the analysis and testing needed. (Clear Abstractions, Reduced Complexity, and Structured Decomposition and Composition)
ALWAYS INVOKED	The protection provided by a mechanism or feature that is not always invoked is not continuous, and, therefore, a loss may occur while the mechanism or feature is suspended or turned off. (Continuous Protection)
TAMPER-PROOF	The mechanism or feature and the data that the mechanism or feature depends on cannot be modified in an unauthorized manner.

The design criteria described above are based on the generalized reference monitor concept. The reference monitor concept[72] is an abstract model of the necessary and sufficient properties that must be achieved by any mechanism that performs an access mediation control function [21] [37]. The reference monitor concept is a foundational access control concept for assured system design. It is defined as a trustworthy abstract machine that mediates all accesses to objects by subjects [38]. As a concept for an abstract machine, the reference monitor does not address any specific implementation. A reference validation mechanism, which includes a combination of hardware and software, realizes the reference monitor concept to provide the access mediation foundation for a trustworthy secure system.

The generalized reference monitor concept and the four essential design criteria can be used effectively as the design basis for individual system elements, collections of elements, networks, and systems where intentional and unintentional adversity can prevent the realization of a loss. The reference monitor concept drives the need for rigor in engineering activities commensurate with the trust to be placed in the system or its constituent system elements.[73] The concept describes an abstract model of the necessary properties that must be realized by any mechanism that claims to achieve a constraint or set of constraints and the basis for determining the extent to which the properties are satisfied. A mechanism that achieves successful constraint has two parts: (1) a means to decide whether to constrain or not constrain and (2) enforcement of the decision. Enforcement of the decision must sufficiently:

- Enforce constraints to achieve only the authorized and intended system behaviors and outcomes
- Provide self-protection against targeted attacks on the mechanism enforcing the decision (including applying the essential design criteria)

[72] The reference monitor concept is described in the *Trustworthy System Control* principle in Appendix E.
[73] Conceptually, the reference monitor concept can be extended to any control function that is to enforce a system constraint [39].

- Be absent of self-induced, emergent, erroneous, unsafe, and non-assured control actions

The protection characteristics for mechanisms must account for but not depend on having detailed knowledge of the capability, means, and methods of an adversary.

D.4.3. Security Function Failure Analysis

The design principle of *Protective Failure* states that a failure of a particular system element should neither result in an unacceptable loss nor invoke another loss scenario. The failure of a security function is of special concern, given the need for security functions to always be invoked and operating correctly. Consequently, failure analyses must be performed during system design to determine the impacts of security function failure on the system capabilities.

Failure analyses consider the assets that may be impacted by security function failure and the associated loss consequences. Failure analyses also consider the function allocation to system elements and the way the system function and element combination interacts with other system function and element combinations, independent of specific events and conditions that might lead to the failure. The principles for trustworthy secure design in Appendix E serve to guide and inform the analyses.

The outcomes of the security function failure analyses also drive assurance levels and objectives, as well as the fidelity and rigor of architecture, design, and implementation methods employed to achieve those objectives. Assurance considerations are discussed in Appendix F.

THE SCIENCE BEHIND SECURITY

"Each of these [design] requirements [for mechanisms] is significant, for without them, the mechanism cannot be considered secure. The [need to be tamper-proof] is obvious, since if the reference validation mechanism can be tampered with, its validity is destroyed, as is any hope of achieving security through it. The [third] requirement of always invoking the reference validation mechanism simply states that if the reference validation is (or must be) suspended for some group of programs, then those programs must be considered part of the security apparatus and be [tamper-proof and evaluatable]. The [evaluatable] requirement is equally important. It states that because the reference validation mechanism is the security mechanism in the system, it must be possible to ascertain that it works correctly in all cases and is always invoked. If this cannot be achieved, then there is no way to know that the reference validation correctly takes place in all cases, and therefore there is no basis for certifying a system as secure."

-- James P. Anderson
The Anderson Report [37]

D.4.4. Situational Awareness

Situational awareness is a foundational security means objective. That is, to achieve other security objectives, situational awareness is necessary and must be accounted for in design. For example:

- Mediating access requires situational awareness in cases where rules for granting access involve timing, sequence, state, and other conditions about the system and prior access.
- Preventing and limiting loss are informed by comprehensive data and information about system states and conditions (*Anomaly Detection*).

Situational awareness requires the ability to accurately detect, capture, record, and analyze the needed characteristics and details of the system's behaviors and actions at a frequency and with the granularity necessary to act and/or inform external entities for subsequent action to be taken.[74] False positives and false negatives are to be avoided to the extent practicable.

Given the potential consequences of compromises of situational awareness capabilities and wrongful attribution, the mechanisms used must meet the essential design criteria (Section D.4.2) with the appropriate rigor. The system audit logs and other system records often need stringent protection, such as using *Distributed Privilege* for access and storing the logs and records in a separate subsystem (*Domain Separation*).

D.4.5. Trade Space Considerations

System design involves trade space decisions. Decision-making about protecting assets is guided by a determination of valuation that informs asset criticality and priority (e.g., assessing the positive effect in achieving objectives and the negative effect for any loss associated with an asset). The criticality and priority based on valuation are used in investment decisions on the type, rigor, and expected effectiveness of protection (*Commensurate Protection*). Decisions may also be guided by the costs and benefits from different design options.

The costs associated with a trustworthy secure design approach include the cost to acquire, develop, integrate, operate, and sustain the security features; the cost of the security features and functions in terms of their system performance impact; the cost of security services used by the system; the cost of developing and managing life cycle documentation and training; and the cost of obtaining and maintaining the target level of assurance.

The cost of analysis to substantiate the trustworthiness claims of certain design choices is also an important trade space factor. Given two equally effective design options, the more attractive of the two options may be the one that has a lower relative cost to obtain the assurance needed to demonstrate satisfaction of trustworthiness claims. In all cases, assess the cost of system security at the system level; and consider trustworthiness objectives and the cost that is driven by the assurance activities necessary to achieve the trustworthiness objectives. Trustworthiness design

[74] Common organizational actions include (1) responses to security-relevant anomalies, such as remedial training for users or replacing the right system component responsible for undesired system behaviors, and (2) audits of system activities, including assessing for suspicious patterns of access that indicate insider threats and to satisfy accountability regulations, such as those required of financial institutions.

principles such as *Commensurate Rigor* and *Commensurate Trustworthiness* inform the trade space analysis.

The benefits of a design option are determined by its effectiveness in providing the required protection capability, the trustworthiness that can be placed on it, and the loss potential associated with it, given the value, criticality, exposure, and importance of the assets protected. An optimal balance between cost and benefit may be realized by using a less costly combination of engineering activities and system features and functions rather than the use of a single cost-prohibitive activity or security feature or function. Moreover, an adverse performance impact may preclude some security options.

CONSERVATION OF RISK

"The law of conservation of energy states that energy can neither be created nor destroyed, only change in form. This law has many important implications in engineering, including implying the impossibility of creating a perpetual motion machine … There is a parallel pseudo-principle that is often offered in a half-joking maxim – risk can neither be created nor destroyed, only moved around. It is not universally true [but] it is worthwhile considering the pseudo-principle because often, a change to a design often does end up 'squeezing the risk balloon,' only to discover that the risk appears elsewhere, perhaps in an unexpected place in the system, which could cause the defended system to be less secure than the engineering intended."

-- O. Sami Saydjari
Engineering Trustworthy Secure Systems [62]

Appendix E. Principles for Trustworthy Secure Design

This appendix describes the design principles that serve as the foundation for engineering trustworthy secure systems.[75] The principles provide a basis for reasoning about a system. They should not be applied as *rules* to be complied with, nor should they be prioritized, sequenced, or ordered for prescriptive application or used as a basis for making judgments on conformance.

The use of the design principles is subject to various priorities and constraints that may restrict or preclude their application within a specific context.[76] The principles may conflict with other principles, and that conflict must be understood. In practice, the principles can be satisfied or implemented in various and often equally effective ways. The use of specific principles may change in response to changes and variances in requirements, architecture, design, and risk acceptability. Therefore, their application should be planned for, appropriately scoped, and revisited throughout the engineering effort.

KEY SECURITY OBJECTIVE

An important objective for security is the reduction of uncertainty regarding the occurrence and effects of adverse events. Reducing the uncertainty of adverse events is achieved by eliminating hazards, susceptibility, and vulnerability to the extent possible. Where elimination cannot occur, their effects should be controlled to the extent possible. Applying the design principles for trustworthy secure systems is a means of achieving the elimination and control of the hazards, susceptibility, and vulnerability that lead to adverse events [39].

The principles for trustworthy secure design are representative of the practices of the safety, security, reliability, survivability, and resilience communities and the specialty engineering disciplines associated with those communities. Collectively, the goals of these practices represent the *end objectives* that the system must satisfy for trustworthy control of adverse effects. The principles are grounded in research, development, and application experience starting with the early incorporation of mechanisms into trusted operating systems to today's components, environments, and systems and are expected to remain universally applicable for new, emerging, and maturing approaches. The concepts and theorems from the disciplines of computer science, computer engineering, systems engineering, control systems, fault/failure tolerance, software engineering, and mathematics – as employed across the communities and specialties – constitute the means to achieve the end objectives.[77] The principles for trustworthy secure design are listed in Table 4.

[75] NIST acknowledges the significant contributions of the Naval Postgraduate School Center for Information Systems Security Studies and Research and The MITRE Corporation in providing content for this appendix. The content was informed by the research reports of the principal investigators from those organizations [21] [39].

[76] Engineering judgment considerations for the application of the principles for trustworthy secure systems are described in [39].

[77] For example, trustworthiness requires that mechanisms be evaluatable (Section D.4.2). Consequently, many principles deal with reducing and managing complexity and creating systems that can be more easily evaluated. See [41] for discussions on how systems may be too complex to be analyzed for adequate assurance.

Table 4. Principles for Trustworthy Secure Design

DESIGN PRINCIPLE	DESIGN PRINCIPLE
Anomaly Detection	Least Privilege
Clear Abstractions	Least Sharing
Commensurate Protection	Loss Margins
Commensurate Response	Mediated Access
Commensurate Rigor	Minimal Trusted Elements
Commensurate Trustworthiness	Minimize Detectability
Compositional Trustworthiness	Protective Defaults
Continuous Protection	Protective Failure
Defense in Depth	Protective Recovery
Distributed Privilege	Reduced Complexity
Diversity (Dynamicity)	Redundancy
Domain Separation	Self-Reliant Trustworthiness
Hierarchical Protection	Structured Decomposition and Composition
Least Functionality	Substantiated Trustworthiness
Least Persistence	Trustworthy System Control

E.1. Anomaly Detection

Principle: Any salient anomaly in the system or its environment is detected in a timely manner that enables effective response action.

Note: The purpose of anomaly detection is to identify the need to take corrective action to address a loss condition that has occurred or that will occur if conditions that affect the system behavior are allowed to persist. Anomaly detection is critical to achieving loss control objectives to prevent and limit loss and its adverse effects. The detection of such anomalies requires monitoring system behaviors and outcomes to determine when any deviations from the design intent occur. It also requires monitoring conditions in the environment to identify or forecast those conditions that can cause an anomaly in the system if corrective action is not taken.

The "timely manner" aspect of anomaly detection reflects the urgency to detect emerging loss conditions as early as possible. Early detection increases response action options, such as graduated response options, and ensures that response actions have sufficient time to have an effect. When the determination of response involves humans in the loop, early detection enables a more reasoned judgment of proper response.

Anomaly detection can be implemented at varying levels of abstraction (e.g., system, sub-system, assembly, function, mechanism) and may occur in periodic, aperiodic, or event-driven manners. The basis for anomaly detection within the system is the expectation that the system behaviors, outcomes, and interactions produced are expected to remain consistent, adhere to some norm, or are deterministic across all system states and modes. The types of anomalies include those associated with the results of system behavior; state consistency; continuity of function; integrity, correctness, and trustworthiness of system elements; system configuration; and the abuse or misuse of the system.

The basis for anomaly detection in the environment differs from that in the system because the environment is not under the control of the system. The environment presents a wide range of adversity to the system, and the system is designed to achieve its design intent within defined bounds of

environmental conditions. Those bounds can be treated as the "norm" for anomaly detection, whereby environmental conditions that are trending beyond the norm or that reflect conditions outside of the norm may result in an adverse effect on the system, thus requiring a planned response to prepare for an impending difficulty or crisis.

Anomaly detection requires capturing data to support all intended response actions for a detected anomaly, including attribution-related data. Consequently, the fidelity in data describing the anomaly must be commensurate with the consequences of the loss scenarios associated with the anomaly and of wrong responses in addressing the detected anomaly. The responses taken will often rely on attribution to uniquely identifiable entities that may be responsible for undesired actions, behaviors, or outcomes. For non-human entities, corrective actions may include component replacements, repairs, or other corrections. For human entities, these may include training, remediation, or disciplinary actions. Wrongful attribution may have undesired consequences, such as the cost of unnecessarily repairing the wrong system element while an undesired condition persists or the wrongful termination of an individual. Attribution rigor is driven by the needed proof that an entity is responsible for an anomaly.

Three aspects of anomaly detection are necessary to provide criteria for an appropriate response action or set of actions:

- *Basis for Correctness:* A system model provides a basis against which actual behavior and outcomes can be compared to confidently enable conclusions that an anomaly exists or to determine or forecast that an anomaly is about to occur. System models include normal, contingency, degraded, and other system states/modes of operation and account for the adversity to which the system is subjected.
- *Data Collection:* Systems capture self-awareness data in the form of health, status, test, and other data indicative of actual behavior and outcomes, including traceability to support attribution. Terms for data collection include instrumentation, monitoring, logging, auditing, self-tests, and built-in tests.
- *Data Interpretation:* The interpretation of data allows for conclusions of unacceptable or suspicious events that have happened (e.g., halt or failure condition), that are progressing (e.g., approaching a threshold of failure condition), or that can be expected to happen (i.e., in the absence of change, the failure condition will occur), including tracing to responsible entities to inform appropriate responses to events.

Caution must be taken with the use of design features that may hinder anomaly detection. Poorly designed lines of defense for defense in depth have been found to conceal emerging dangerous system states and conditions, especially from human observers [40]. The system design must minimize the difference between estimated system states and conditions and actual system states and conditions.

Two approaches to anomaly detection are:

- *Self-Anomaly Detection:* An entity has no dependency on another entity to detect an anomaly within the scope of its intended design. Self-anomaly detection usually involves an axiomatic or environmentally enforced assumption about its integrity. Typically, trusted elements have the capability for self-anomaly detection. This means that at the highest level of trustworthiness, an entity must be able to assess its internal state and functionality to a meaningful extent at various stages of execution. The detected anomalies must correlate to the trustworthiness assumptions placed on the entity.
- *Dependent Anomaly Detection:* An entity-of-interest is dependent on another entity for some or all anomalies that are detected. When an entity-of-interest relies on another entity for any portion of the assessment, that entity must be at least as trustworthy as the entity-of-interest.

References: [20] [40] [43]

E.2. Clear Abstractions

Principle: The abstractions used to characterize the system are simple, well-defined, accurate, precise, necessary, and sufficient.

Note: Abstractions can help manage the complexity of the system [24]. Clarity in the abstract representations of the system facilitates an accurate understanding of the system and how the system functions to deliver the required capability. Clear abstractions also reduce the potential for misunderstanding or misinterpretation of what is represented by the abstraction. Applying the principle of clear abstractions means that a system has simple, well-defined interfaces and functions that provide a consistent and intuitive view of the data and how it is managed. The elegance (e.g., accuracy, precision, simplicity, necessity, sufficiency) of the system interfaces – combined with a precise definition of the functional behavior of the interfaces – promotes ease of analysis, inspection, and testing, as well as the correct and secure use of the system. Examples that reflect the application of this principle include avoidance of redundant, unused interfaces; information hiding;[78] and avoidance of semantic overloading of interfaces or their parameters (e.g., not using one function to provide different functionality, depending on how it is used).

It is important to ensure that the proper rigor is applied in the development of system abstractions during design. Clarity in the abstract representation of the system requires the use of well-defined syntax and semantics with elaboration as needed to ensure that the representations are well-defined, precise, necessary, and sufficient. Clear abstractions promote confidence in analysis, verification, and the correct use of the system. Abstractions can be achieved using models, including Systems Modeling Languages.

References: [2] [20] [21] [24]

E.3. Commensurate Protection

Principle: The strength and type of protection provided to a system element are commensurate with the most significant adverse effect that results from a failure of that element.

Note: The strength and effectiveness of the protection for a system element must be proportional to the need. As the need increases, the protection of that element should also increase to the same degree. Need is derived from the most significant adverse effect associated with the system element or the trust that is placed in the element. The protection can come in the form of the system element's own self-protection, from protections provided by the system architecture, or from protection provided by other elements. The needed strength of protection is independent of these design choices (or others, such as distributed versus centralized design), a concept sometimes referred to as secure distributed composition [2]. Furthermore, confidence in the effectiveness of the protections provided to a system element should also increase commensurate to the need. This is addressed by the principle of *Commensurate Rigor*.

References: [2] [21]

E.4. Commensurate Response

Principle: The system design matches the aggressiveness of an engineered response action's effect to the needed immediacy to control the effects of each loss scenario.

[78] The term *information hiding*, also called representation-independent programming, is a design discipline to ensure that the internal representation of information in one system component is not visible to another system component invoking or calling the first component, such that the published abstraction is not influenced by how the data may be managed internally.

Note: The selected response to a detected anomaly considers three factors to determine the effect that the response has on the loss and the system:

- The effectiveness and aggressiveness of the engineered response to directly address the anomaly and to prevent or limit the loss
- The direct, residual, or side effect of the response on the system
- The opportunities that remain to take other response actions should the selected response fail to achieve the intended result

Responses can be achieved by a combination of manual, semi-automated, fully automated, or autonomous means. However, the response action is distinct from the determination that a response is necessary and from the notification or signaling that invokes the response action.

Commensurate responses require consideration of the response-effect-consequence relationship associated with a specific loss. Ideally, for any given need for a response, a single action taken will be effective to resolve the loss concern and will have no associated adverse effect. Practically, due to complexity and the limits of certainty, the response action may not have the desired effect, may compound the problem, or may cause another problem. The balance required is one that determines if, when, and how a response action should be taken to be initially more aggressive or initially less aggressive. The severity of the problem and the time available for an effective response dictate a strategy for a continuum of responses, characterized by two extremes:

- *Graduated Response:* A graduated response is initially the least aggressive or impactful action possible to prevent the loss from continuing or escalating and does so with consideration of the possible side effects associated with the response action. The graduated response allows for taking increasingly more aggressive action should the loss situation persist or escalate.
- *Ungraduated Response:* An ungraduated response is the most aggressive and impactful action to prevent the loss from continuing or escalating and does so without consideration of the potential side effects associated with the response action. The ungraduated response recognizes the severity of the loss as justifying the most aggressive action, even if that option provides no alternatives should it fail to have the intended or desired effect or if it causes other losses to occur.

Without early observability of potential loss, the option for a graduated response may not exist. A commensurate response is aided by early detection, which in turn increases the options for a graduated response.

References: [40]

E.5. Commensurate Rigor

Principle: The rigor associated with the conduct of an engineering activity provides the confidence required to address the most significant adverse effect that can occur.

Note: Rigor determines the scope, depth, and detail of an engineering activity. Rigor is a means of providing confidence in the results of a completed engineering activity. Generally, an increase in rigor translates to an increase in confidence in the results of the activity. Further, increased confidence reduces the uncertainty that can also reduce risk or provide a better understanding of what to address to achieve risk reduction. The relationship between rigor and the criticality of data and information used to make decisions is recognized by systems analysis practices [4].

The principle of commensurate rigor helps to ensure that the concept of rigor is included as an equal factor in the trade space of capability, adverse effect, cost, and schedule in the planning and conduct of engineering activities, method and tool selection, and personnel selection. An increase in rigor may

translate into an increase in the cost of personnel, methods, and tools required to complete rigorous engineering activities or an increase in schedule to accomplish the activities with the expected rigor. Any increased cost that may occur can be justified by acquiring confidence about system performance to limit loss while also addressing the system's ability to deliver the capability. Therefore, the rigor associated with an engineering activity should be commensurate with the significance of the most adverse effect associated with the activity.

References: [2] [4]

E.6. Commensurate Trustworthiness

Principle: A system element is trustworthy to a level commensurate with the most significant adverse effect that results from a failure of that element.

Note: A trusted element continuously exhibits properties of trust during the time that it is depended upon by other system elements. The degree of trustworthiness needed for a trusted element is determined by those entities that depend on the element. Some basis is required to support decisions about trust and trustworthiness. The basis includes expressing the trust that is to be placed in a system element, expressing the trustworthiness that is exhibited by the element, and comparing the trustworthiness of different system elements. This principle is particularly relevant when considering systems and elements with complex chains of trust dependencies.

References: [4] [20]

E.7. Compositional Trustworthiness

Principle: The system design is trustworthy for each aggregate composition of interacting system elements.

Note: The trustworthiness of an aggregate of composed system elements cannot be assumed based on the trustworthiness assertions of each individual element in the aggregate. Further, the trustworthiness of an aggregate of composed trustworthy system elements cannot be assumed to be equal to the trustworthiness of the least trustworthy element in the aggregate. By definition, a system is a combination of interacting system elements. Each system function results from the emergent behavior of a composed set of system elements. Similarly, the trustworthiness of a composed set of system elements is an emergent property of the composition. Therefore, the trustworthiness of the composed set of system elements (i.e., aggregate) for a given system function must be determined by treating the aggregate as a single discrete element. The compositional trustworthiness principle addresses how an argument can be made for system-level trustworthiness given how the constituent elements of the system compose to form the system and do so by adhering to the composition principles.

References: [2] [4] [36] [42]

E.8. Continuous Protection

Principle: The protection provided for a system element must be effective and uninterrupted during the time that the protection is required.

Note: The protection capability must be uninterrupted across all relevant system states, modes, and transitions for there to be assurance that the system can be effective in delivering the required capability while controlling loss. Continuous protection requires adherence to the following principles:

- *Trustworthy System Control:* Every controlled action is constrained by the mechanism, and the mechanism can protect itself from tampering. Sufficient assurance of the correctness and completeness of the mechanism can be ascertained from analysis and testing.
- *Protective Failure* and *Protective Recovery:* A protective state is preserved during error, fault, failure, and successful attack, as well as during the recovery of assets or of recovery to normal, degraded, or alternative operational modes.

Continuous protection applies to all configurations, states, and modes of the system, as well as the transitions between those configurations, states, and modes. The system design must ensure that protections are coordinated and composed in a non-conflicting and mutually supportive manner across the non-behavioral aspects of the system structure and the behavioral aspects of system function and data flow.

While the design for continuous protection applies for the entire time that the protection is required, the protection capability is sometimes intentionally disabled (e.g., Battleshort[79] intentional override). The intentional disabling/override of protection is an exception case and, therefore, does not violate this principle. That is, the principle of continuous protection applies only for the entirety of time that the protection is required and not knowingly and intentionally disabled.[80]

References: [21]

E.9. Defense In Depth

Principle: Loss is prevented or minimized by employing multiple coordinated mechanisms.

Note: The coordinated deployment of multiple protective mechanisms for a system helps to avoid single points of failure. The principle of defense in depth has three pillars:

- Multiple lines of defenses or barriers should be placed along loss scenario sequences.
- Loss control should not rely on a single defensive element.
- The successive barriers should be diverse in nature and include technical, operational, and organizational barriers [52].

Defense in depth requires the use of coordinated mechanisms (active) within an architectural structure (passive) that achieves the depth characteristic.[81] Ideally, the initial lines of defense prevent loss, while subsequent lines of defense block loss scenario escalation and/or contain loss and potential consequences when needed. A defense-in-depth strategy examines loss scenarios for those points of opportunity to prevent or contain loss. It also leverages the opportunities to use active or passive mechanisms or constraints to meet loss control objectives.

The coordination of defense-in-depth mechanisms (i.e., combinations of structural, data, and control flow coordination) in conjunction with other design principles (e.g., *Anomaly Detection*, *Commensurate Response*) reflects a design strategy to satisfy the specified loss control objectives.

While defense in depth distributes the protection capability to many components, a defense-in-depth strategy may also consider a distributed composition to a line of defense. A protection capability provided by a single system component is a potential single point of failure or bottleneck to system performance. It

[79] Battleshort is the capability to bypass normal interlocks in mission-critical equipment (e.g., equipment that must not be shut down or the mission function will fail) during battle conditions [51].

[80] However, the inclusion of a capability for intentionally disabling/overriding protection requires additional control features, devices, and associated analysis for the enforcement of constraints to prevent the inadvertent actuation of the override capability.

[81] While the discussion in this section is limited to the machine, defense in depth may involve a combination of technical, operational, and organizational elements. Additional discussion on defense in depth can be found in [52].

may also raise other concerns. A distributed composition of a defense layer may provide additional options within the coordination of layers.

Defense in depth is, in part, a form of the principle of *Protective Failure*. It helps satisfy the objective that a failure of a system element should not result in an unacceptable loss. However, it does not satisfy the objective that a failure of a system element should not invoke another loss scenario.

References: [2] [21] [40] [47]

E.10. Distributed Privilege

Principle: Multiple authorized entities act in a coordinated manner before an operation on the system is allowed to occur.

Note: Distributed privilege[82] is a means to require agreement and coordination from multiple entities when performing an operation, thereby preventing a single entity from acting alone. Distributed privilege separates, divides, or in some other manner distributes the privileges required to perform an operation among multiple entities. This distribution includes a set of rules, conditions, and constraints that describe how multiple entities must interact through positive actions before a requested operation can proceed and be completed. The rules, conditions, and constraints may reflect combinations of the following:

- *Simultaneous Actions:* Multiple different authorized entities execute a command within a specified time window.
- *Sequenced Actions:* Multiple different entities interact within a linear sequence of actions where each successive action is enabled only by the successful completion of a prior action.
- *Parallel Actions:* Multiple entities execute sequences concurrently, and success is achieved either by a consensus of the results of each concurrent action or by voting among the participants.

Defeating distributed privilege requires collusion to take an unauthorized or improper action. In the case of an attack, distributed privilege forces the adversary to target all of the entities to whom privilege is distributed.

References: [21] [46]

E.11. Diversity (Dynamicity)

Principle: The system design delivers the required capability through structural, behavioral, or data or control flow variation.

Note: A system design that incorporates diversity helps to avoid common mode failures and introduces unpredictability to adversaries, thus complicating the planning and execution of where, when, and how to target their attacks. While the system behaviors that result from a design may be unpredictable from the viewpoint of the adversary, the design itself must be predictable and verifiable in achieving only the intended outcomes. The options for diversity include variety in the system structural and architectural design elements, the system functional and behavioral elements, the interfaces and interconnections between interfaces, the data and control flow, and the technology and component selection. Diversity can reside in:

- Fixed or static characteristics of the system (e.g., multiple instances of a system element, multiple communication channels)

[82] Saltzer and Schroeder [46] originally named this the separation of privilege. It is also equivalent to separation of duty.

- Variable or dynamic characteristics of the system (e.g., reconfiguration, relocation, refresh of system elements; random routing of data over different communication channels from source to destination; the ability to change aspects of the system behavior, structure, data, or configuration in a random but nonetheless verifiable manner)

A design approach that includes diversity in structure, configuration, communications, protocols, and similar or dissimilar system elements (e.g., N-version, heterogeneity) increases uncertainty due to the increased complexity of the design and the behaviors and outcomes that stem from emergent effects, side effects, and feature interaction. This drives the need for confidence that the design approach will deliver only the authorized and intended functional behavior, produce only the authorized and intended outcomes, and do so in a manner that allows for control over side effects, emergence, and feature interaction.

Diversity options include intentionally designed regular or irregular changes in the system (e.g., implementing the concept of dynamicity). A design incorporating dynamicity can (1) complicate the attack planning of an adversary, (2) reduce the potential for non-adversarial adversity to have an effect on the system, (3) provide the margin to deliver a required capability while reducing actual losses, and (4) protect against the effects of an attack. Dynamic change may refer to either shifting the target or shifting the behaviors of a target in performing its activities (e.g., frequency hopping complicates attempts to intercept or jam signals within wireless communications).

The uncertainty and diminished predictability associated with the employment of diversity and dynamicity in design can be problematic where it impedes or prevents having confidence that the system will function and produce outcomes only as authorized and intended. It is important to differentiate where the uncertainty lies: (1) uncertainty in how the system achieves an end objective (i.e., the means to an end) or (2) uncertainty that an objective will be achieved (i.e., achieving the end). A design that employs diversity and dynamicity must be based on acquiring confidence that the system will produce only the desired results despite uncertainty in knowing exactly how the desired results are achieved. This constitutes a design trade that is specific to diversity- and dynamicity-based designs. Diversity may have a cost (e.g., hardware, software, maintenance, training, assurance) greater than the value or effectiveness that it provides.

References: [20] [47] [45]

E.12. Domain Separation

Principle: Domains with distinctly different protection needs are physically or logically separated.

Note: The separation of domains enables enhanced control and, therefore, protection of system function and the flow of data. Control relative to separated domains limits the extent to which an entity or domain is influenced by or is able to influence some other entity or domain, thereby enhancing the protection of a domain. This is achieved through the control of information flow and data between domains as well as control over the use of a system capability between domains.

The differing protection needs that are used to define domains may be thought of in terms of protecting the domain from influence by external entities (i.e., susceptibility) and protecting external entities from erroneous behavior that occurs within the domain (i.e., containment). This distinction may include separating critical functions from less critical functions, such as separating the flight control functions of a transport aircraft from the environmental control functions that maintain a safe environment for the cargo and passengers being transported.

Historically, domain separation has been used to enforce the separation of roles or privileges. For example, a system may separate an “administrative” or “supervisor” domain from “user” domains. The

administrative domain is accessible only by system administrators with proper privileges, and distinctly administrative functions may only be executed by administrators from the administrative domain. Similarly, data intended to be accessed only by administrators and administrative functions (e.g., system configurations) is stored and accessed only within that domain, ensuring needed protection of the data.

Domain separation requires a domain to be contained within its own protected subsystem so that elements of the domain are only directly accessible by procedures or functions of the protected subsystem. The concept of isolation enables the implementation of domain separation. Isolation limits the extent to which one domain can influence or can be influenced by other entities. The challenge is that the system elements within domains must at times interact with other elements and the environment to deliver a capability. Every interface that results from design decisions can diminish domain separation while achieving requirements for a system capability. External requests for resources or functions within protected subsystems are arbitrated at these interfaces. Firewall, data diodes, and cross-domain solutions (CDS) are examples of mechanisms that enable varying degrees of control over the interactions between separated domains.

Encryption is another mechanism often used to provide domain separation. For example, communication between distinct subsystems within a domain may be encrypted with a key that is known only to the subsystems within the domain. Where a common storage module or subsystem is used for multiple domains, encryption may be used to limit information access to the domain that owns the key to decrypt.

References: [21] [43]

E.13. Hierarchical Protection

Principle: A system element need not be protected from more trustworthy elements.

Note: Hierarchical protection is a simplifying assumption for trade decisions to help determine where emphasis is placed in providing protection and the extent of the protection effectiveness. The simplifying assumption introduces susceptibilities to system elements that are dependent on more trustworthy elements. The assumption relies on validated trust assertions about the more trustworthy element and acceptable uncertainty associated with behavior outside of the scope of the validated trust assertions. For example, systems may include a human element, which is often the more trustworthy element. The assertions of the trusted human are violated for the malicious insider threat. The extent to which any element is considered trustworthy has limits, and beyond those limits, the element should not be assumed to remain trustworthy. In the degenerate case of the most trustworthy system element, it must protect itself from all other elements. For example, if an operating system kernel is deemed the most trustworthy component in a system, then it must protect itself from the less trustworthy applications it supports. However, the applications do not need to protect themselves from the operating system kernel.

References: [2] [43]

E.14. Least Functionality

Principle: Each system element has the capability to accomplish its required functions but no more.

Note: Susceptibility and vulnerability increase unnecessarily when a system element provides more functionality than is needed to achieve its intended purpose. Least functionality reduces the potential for susceptibility and vulnerability and reduces the scope of analysis of the system element's trustworthiness and loss potential. The strictest interpretation of least functionality is to prohibit any system element functions that are not required. Where that is not possible or practical, the unnecessary functions of the system element should be disabled, disarmed, or put into a "safe" mode that prevents the functions from

being used. In all other cases, mediated access can be used to prevent access to and use of the unneeded functions.

An example of when it may not be possible or practical to avoid unnecessary functions is the use of commercial off-the-shelf (COTS) components. COTS components typically contain functions beyond those required to fulfill their intended purpose. In such cases, the components should be configured to enable only the functions that are required to fulfill their purpose and prohibit or restrict functions that are not required to fulfill their purpose.

References: [2] [21]

E.15. Least Persistence

Principle: System elements and other resources are available, accessible, and able to fulfill their design intent only for the time for which they are needed.

Note: Least persistence reduces susceptibility. It limits the extent to which functions, resources, data, and information remain present, accessible, and usable when not required, thereby reducing the opportunity for their inadvertent or unauthorized use, modification, or activation.

The broadest interpretation of least persistence is to not install, instantiate, or apply power to system elements and resources until needed and to completely remove system elements or power from system elements and resources when they are no longer required. When it is not possible or practical to do so, those system elements and resources should be fully disabled, disarmed, or put into safe mode to prevent their ability to function or be used. Finally, when it is not possible to disable, disarm or put into safe mode, access to those elements and resources should be mediated to constrain the time and duration of their use (*Mediated Access*).

Three conditions must be satisfied for an active system element or resource to be usable, and two of these conditions apply to non-active elements or resources:

- *Presence (active and non-active):* The system element or resource must be installed, loaded, residing in memory (software), and configured.
- *Accessible (active and non-active):* The system element or resource must be invoked, interacted with, or operated on.
- *Able to Function (active):* The system element or resource must be able to execute (i.e., powered on, enabled, or armed) to deliver a service or perform a function.

Least persistence is reflected in concepts such as sanitizing, erasing, and clearing memory and/or storage locations; disabling, removing, and disconnecting network ports, system interfaces, and the services provided by system interfaces; powering off and unplugging hardware when not needed; and instantiating software just before need and de-instantiating after it is no longer needed. Least persistence has added benefits that include simplifying the processes of:

- Cleansing the system element to remove corrupted aspects or side effects
- Re-establishing the system element to a known state (i.e., a refresh)
- Minimizing the time in which system elements are exposed to the environment, to attack, and to erroneous behavior

Where system elements or resources are removed and then restored as needed, there must be a trusted representation of the system element and a trusted ability to instantiate that system element within the time constraints for its use.

References: [53]

E.16. Least Privilege

Principle: Each system element is allocated privileges that are necessary to accomplish its specified functions but no more.

Note: System elements can be implemented by entities such as hardware, firmware, software, and personnel. By design, the system must be able to limit the scope of a system element's actions. This has two desirable effects: (1) the impact of a failure, corruption, or misuse of the element is minimized, and (2) the analysis of the system element is simplified. A design driven by least privilege considerations results in a sufficiently fine granularity of privilege decomposition and the ability for the fine-grained allocation of privileges to human and machine elements.

The application of the principle of least privilege means allocating to a system element only the privileges that are necessary to permit that element to perform the functions required of it. This could include a need to modify, delete, use, or configure a resource, or to authorize, start/enable, or stop/disable a process [20].

Least privilege can inform the use of other principles such as the employment of *Domain Separation* and *Structured Decomposition and Composition*. That is, the system modules can be designed so that only the system elements encapsulated by the module are directly accessed or operated on by the functions within the module, thus aiding the implementation of least privilege.

References: [2] [20] [21] [46]

E.17. Least Sharing

Principle: System resources are shared among system elements only when necessary and among as few elements as possible.[83]

Note: Sharing via common mechanism and other means can increase the susceptibility of system resources (e.g., data, information, system variables, interfaces, functions, services) to unauthorized access, disclosure, use, or modification and can adversely affect the capabilities provided by the system. According to Saltzer and Schroeder [46], "Every shared mechanism (especially one involving shared variables) represents a potential information path between users and must be designed with great care to be sure it does not unintentionally compromise security." A design that employs least sharing helps reduce the adverse consequences that can result from sharing system functions, state, resources, and variables among different system elements. A system element that corrupts a shared state or shared variables has the potential to corrupt other elements whose behavior is dependent on the state. Minimized sharing also helps to simplify the design and implementation [54].

Two criteria provide the basis for applying the principle of least sharing: (1) share only if absolutely necessary, and (2) minimize sharing if allowed. The first criterion is a trade decision that weighs the cost and benefit of sharing resources against the increased exposure that results from the sharing. The second criterion is a constraint on the extent of sharing.

References: [2] [21] [46] [54] [55]

[83] The historically well-known security design principle *least common mechanism* is an instance of least sharing. The principle of least common mechanism is described in [55].

E.18. Loss Margins

Principle: The system is designed to operate in a state space sufficiently distanced from the threshold at which loss occurs.

Note: Margins refer to the difference between a conservative threshold at which the system is expected to operate while subjected to adversity and the point at which the adversity results in failure. Loss margins are created by engineered features put in place to maintain operational conditions and the associated adversity level at some distance (i.e., conservative threshold) from the estimated critical adversity threshold or loss-triggering threshold. Loss margins also allow for increased time to (1) detect the need for a response action (*Anomaly Detection*), (2) determine what the response action should be (*Commensurate Response*), and (3) complete the selected response action. When there is uncertainty about the effectiveness of the response action, loss margins need to allow time to evaluate response effectiveness, determine any additional actions needed, and complete any selected actions.

Uncertainty may derive from the operational environment, the design and realization of the system, the utilization and sustainment of the system, and the adversity presenting itself to the system. Loss margins are effective in addressing uncertainty about how and when a loss-triggering event occurs. Specifically, loss margins are effective in addressing uncertainty associated with:

- Intelligently designed and executed attacks, including attacks that persist and evolve over time
- Unknown, unquantified, and underappreciated susceptibilities, threats, vulnerabilities, hazards, and associated risks

For designs that incorporate loss margins, uncertainty about adversity makes determining the loss-triggering thresholds difficult. Loss margins for design should be determined with a balance between certainty (i.e., what has happened and can happen again) and uncertainty (i.e., what has not happened but can happen, or what has happened but can also happen in a different way). Loss scenarios that include loss escalation and an estimation of the critical threshold for loss occurrence are helpful in making design decisions that incorporate loss margins. Loss scenarios also help to determine the limits of adversity-driven decisions due to uncertainty in knowledge about the adversity (i.e., the adversity is insufficiently known or understood or is just unknown).

Sensitivity analyses must inform the determination of loss margins. Other factors for computing loss margins include system complexity, the use of newer technology or older technology in new ways, and the degree of new environments being introduced. An additional factor is the ability to complete comprehensive and effective testing. Limitations on system test coverage and effectiveness for actual, simulated, or emulated adversity necessitate larger margins to account for the remaining uncertainty. The size of the margin may be reduced over time as unknown and underappreciated loss scenarios are uncovered and corrected. The size of the margin may also need to be increased over time as a malicious adversity capability matures in sophistication.

References: [6] [23] [40] [45] [56] [57] [63]

E.19. Mediated Access

Principle: All access to and operations on system elements are mediated.

Note: Mediated access is a foundational principle in the design of secure systems. The purpose of mediated access is to:

- Place limits on access to and use of the system
- Reduce the possibility of loss escalation

- Reduce the extent to which loss escalates and propagates

Mediated access is based on the interaction between an entity and a target system element and has two aspects:

- *Access to the System Element*: The requesting entity only has authorized access to a target system element.
- *Use of the System Element:* The requesting entity is only allowed to perform authorized operations on the target system element.

Mediated access has two parts: (1) a policy-based access mediation decision and (2) the enforcement of the access mediation decision. The access mediation decision may include conditional constraints that further restrict access (e.g., role, time of day, system state or mode, or duration of operation). If access is not sufficiently mediated, there is no possibility of limiting how system elements (including human and machine elements) interact to ensure that only authorized behaviors and intended outcomes result.

Mediated access is achieved by an access mediation control mechanism. Seminal computer security work has defined the reference validation mechanism as the generalized form of any mechanism that is an implementation of the reference monitor concept (Section D.4.2). The reference monitor provides the design assurance basis for demonstrating the trustworthiness of a mediated access control mechanism. The essential design criteria (Section D.4.2) provide a refinement to extend the generalized reference monitor concept. Mediated access may enforce the constraints described in the principles of *Distributed Privilege*, *Least Privilege*, and *Least Sharing*.

As a predominant security function, mediating access may result in performance bottlenecks if not designed and implemented correctly. The use of a least common mechanism is one means to help reduce bottlenecks, an approach referred to as *efficiently mediated access* [21].

References: [2] [21] [37] [40] [46] [58]

E.20. Minimal Trusted Elements

Principle: A system has as few trusted system elements as practicable.

Note: Minimizing trusted system elements is a cost-benefit trade space consideration employed for the functional allocation of trust within the system. The need for trust is tied to the function provided by a system element, and that need is independent of any distribution of trust across multiple elements in the architecture. The trade decision is, therefore, how best to allocate trust to system elements given the functions they provide and how the elements are best distributed throughout the architecture where distribution is a justified need. Minimizing trusted system elements is one consideration in making that decision.

Trusted elements are generally costlier to construct due to increased rigor in engineering processes and activities. They also require more analysis to qualify their trustworthiness. Minimizing the number of trusted system elements reduces the cost of analysis (i.e., decreases the size, scope, and complexity of the analysis). When the minimization of trusted system elements considers the principle of *Commensurate Protection*, the cost-effectiveness of the analysis is also ensured (i.e., cost of the analysis is justified by the extent of trust required).

Historically, the analysis of interactions between trusted system elements and untrusted system elements is one of the most important aspects of the trust-based verification of system security performance. If these interactions are unnecessarily complex, the security of the system will also be more difficult to ascertain than one whose internal trust relationships are simple and elegantly constructed. In general, fewer trusted components will result in fewer internal trust relationships and a simpler system.

References: [2] [20] [43] [44]

E.21. Minimize Detectability

Principle: The design of the system minimizes the detectability of the system as much as practicable.

Note: A system that is not discoverable, observable, or trackable by an adversarial threat or exposed to such a threat is less prone to a targeted attack. Minimizing detectability drives engineering design decisions to eliminate or reduce exposures such as unnecessary interfaces, access points, footprints, and emanations, thereby reducing susceptibility to adversarial threat actions. Interfaces and access points have the effect of exposing the system to intentional adversity (i.e., attacks) and unintentional adversity (i.e., faults, errors, incidents, accidents). Yet interfaces and access points are necessary to compose system elements to deliver required capabilities, and duplicating interfaces and access points is needed to avoid single points of failure. System design must balance the need for interfaces with the susceptibility that results from the interface being exposed, discovered, and observed. Every interface, whether internal or external, constitutes an exposure that must be considered.

Minimizing detectability reduces the ability of an adversary to observe and discover information about the system to craft and execute attacks. This includes detecting a system's location, presence, and movement (e.g., due to emissions, signatures, or footprints). Ways by which a system may be detected include heat emission, electronic magnetic (EM) emissions, sound, vibrations, reflecting radar waves or light, response to stimulus (e.g., a response to an Internet Control Message Protocol [ICMP] echo request or "ping"), and software traces and thrown exceptions. Specific forms or means to minimize detectability include camouflage, stealth, low probability of intercept/low probability of detect (LPI/LPD) waveforms (for radios), and frequency hopping.

References: [53] [59] [60]

E.22. Protective Defaults

Principle: The default configuration of the system provides maximum protection effectiveness.

Note: The configuration of the system includes the parameters for system functions, data, interfaces, and resources that determine how the system behaves and the outcomes it produces. Protective defaults guarantee that the "as shipped" system configuration and parameters prioritize the achievement of loss control objectives over the ability to deliver a required system capability and performance without dependence on human intervention. Protective defaults require conscientious action to establish the system configuration and parameters that deliver the required capability and performance in a manner that provides *Commensurate Protection* against loss. Protective default configurations for systems include constituent subsystems, components, and mechanisms. The principles of *Protective Failure*, *Protective Recovery*, and *Continuous Protection* parallel this principle to provide the ability to detect and recover from failure.

References: [2] [21] [46]

E.23. Protective Failure

Principle: A failure of a system element neither results in an unacceptable loss nor invokes another loss scenario.

Note: Protective failure, a generalization of the concepts of fail secure and fail safe, is the aspect of continuous protection that ensures that a protection capability is not interrupted during a failure and that the effect of the failure is constrained. Two aspects of protective failure must be satisfied to achieve the intended effect:

- *Avoid Single Points of Failure:* The failure of a single system element should not lead to unacceptable loss. Unacceptable loss should only occur in the case of multiple independent malfunctions – a safety principle known as single failure criterion. The principle of *Defense in Depth* can help achieve this aspect of protective failure.
- *Avoid Propagation of New Failure:* If unmitigated, failures in the system can result in propagating, cascading, or rippling effects on the system. These effects can be addressed if the remaining protections remain effective to prevent the originating failure from causing additional failures.

Protective failure applies to discrete system elements, aggregates of system elements, and systems abstraction. Protective failure seeks to limit a failure's effect to the extent practicable and, in doing so, minimize introducing new loss possibilities. Protective failure can limit the extent to which a failure is able to advance loss scenarios associated with the failure, including cascading losses; trigger a different loss scenario; or create a new loss scenario. Efforts to avoid or limit failures may themselves degrade system performance, which is a form of failure. Thus, system designers may need to consider trade spaces between possible adverse effects and system performance.

References: [2] [21] [40] [47] [45]

E.24. Protective Recovery

Principle: The recovery of a system element does not result in nor lead to unacceptable loss.

Note: Protective recovery is an aspect of *Continuous Protection* that ensures that a protection capability is not interrupted during recovery from actual or impending failure. Protective recovery is applied to discrete system elements, aggregates of system elements, and the system. To the extent practicable, any recovery from impending or actual failure to resume normal, degraded, contingency or alternative operation, or the recovery of other asset losses should not (1) advance the loss scenario that is the target of the recovery, (2) trigger other loss scenarios, or (3) create new loss scenarios. The practicable aspect of this principle recognizes that for some recovery efforts to be successful, they may degrade system performance, which is a form of loss. Protective recovery is an aspect of the response strategy for the system. Thus, graduated and ungraduated considerations of *Commensurate Response* apply to best suit expediency in the need for a protective recovery.

References: [2] [6] [20] [21]

E.25. Reduced Complexity

Principle: The system design is as simple as practicable.

Note: Engineered systems are often complex. Some degree of complexity in the system design is inherent, unavoidable, and must be accepted. The objective of this principle is to ensure that the design reflects the extent to which complexity can be reasonably minimized (i.e., avoid unnecessary complexity).

Complexity can be found in the system structure, interfaces, dependencies, data and control flows, and the interaction of the system with its operational environment. Complexity derives from how the system is decomposed into its individual and aggregates of constituent elements (e.g., subsystems, assemblies), and from how those elements compose through their behaviors and interactions to comprise the functional system.

A more complex design increases uncertainty. Such uncertainty leads to errors and hinders acquiring confidence in the understanding of the design. A complex design is also more prone to erroneous interpretation when conducting the engineering activities of analysis, implementation, and verification throughout the system life cycle [45]. Thus, reduced design complexity contributes to confidence in the technical understanding of the design, enabling more informed trade decisions and decreasing uncertainty in the design and the subsequent realization of the design as a system.

Complexity also increases the difficulty in identifying and assessing loss scenarios, susceptibilities, and vulnerabilities. Conclusions about the nature and effect of vulnerabilities can be reached with a higher degree of assurance in cases of reduced design complexity in contrast to cases where the design is overly complex.

The principle of reduced complexity may also be referred to as the principle of simplification or least common mechanism.

References: [2] [40] [45] [46] [47]

E.26. Redundancy

Principle: The system design delivers the required capability by replication of system functions or elements.

Note: Redundancy employs multiples of the same system elements, data and control flows, or paths to avoid single points of failure. Redundancy requires a strategy for how multiple system elements are used individually or in combination (e.g., load-balancing, fail-over, concurrently, backup, voting, agreement, consensus).

Redundant solutions are susceptible to common mode failure (i.e., a single event that results in the same or equivalent elements failing in the same manner). *Diversity* is a means to address the concerns of common mode failure.

References: [2] [20] [45] [47]

E.27. Self-Reliant Trustworthiness

Principle: The trustworthiness of a system element is achieved with minimal dependence on other elements.

Note: In the ideal case, the trustworthiness of a system element occurs when the claim of trustworthiness is not dependent on protection from another system element. If an element is dependent on other elements to satisfy its trustworthiness claims, then that element's trustworthiness is susceptible to any loss or degradation of the protection capability provided by the other element. The considerations for the extent to which a system element exhibits self-reliant trustworthiness include:

- The trustworthiness objective for the capability
- The trustworthiness of the system element in providing the capability
- The extent to which the capability provided by a system element is dependent on another element
- The extent to which the trustworthiness associated with a capability is dependent on another system element

An argument for self-reliant trustworthiness can be applied at the discrete system element level, at the level of an aggregate of elements, at the system level, or at the system of systems level. In all cases, the distinction between the capability provided and the trustworthiness responsibility for that capability must

be preserved (e.g., self-reliant trustworthiness cannot be claimed if the protection assertions for trust are allocated to and dependent on some other entity). Similarly, when a system capability is distributed across multiple system elements, self-reliant trustworthiness requires that the trust expectations for the capability be properly allocated across the elements that comprise the distributed capability.

The judgment that a system element is self-reliantly trustworthy is based on the element's ability to satisfy a specific set of requirements and associated assumptions. An element that is self-reliantly trustworthy for one set of requirements and assumptions is not necessarily self-reliantly trustworthy for other sets of requirements and assumptions. Any change in the requirement, the satisfaction of the requirement, or in the assumptions associated with the requirement requires reassessment to determine that the element remains self-reliantly trustworthy.

References: [2]

E.28. Structured Decomposition and Composition

Principle: System complexity is managed through the structured decomposition of the system and the structured composition of the constituent elements to deliver the required capability.

Note: The structured decomposition of the system and the subsequent composition of the system elements are guided and informed by the concepts of modularity, layering, and partially ordered dependencies.

Modularity is the system design technique to *divide and conquer* – that is, sub-divide the system into smaller, well-defined cohesive components and assemblies that are referred to as modules. Modularity serves to isolate functions and data structures into well-defined logical units. Modular decomposition can include the allocation of policies to systems in a network, the allocation of system policies to layers, the separation of system applications into processes with distinct address spaces, and the separation of processes into subjects with distinct privileges based on hardware-supported privilege domains. Modular design may also extend to consider trust, trustworthiness, privilege, and policy.

Layering is the grouping of modules into a relational structure with well-defined interfaces, function, data, and control flow so that the dependencies graph among layers is linearly or partially ordered such that higher layers are dependent only on lower layers [2]. Partially ordered dependencies among modules (e.g., if module A depends on module B, then module B cannot depend on module A) and system layering contribute significantly to system design simplicity and coherence. While a partial ordering of all functions and processes may not be possible, the inherent problems of circularity can be more easily managed if the circular dependencies are constrained to occur within layers and minimized within each layer. Partially ordered dependencies also facilitate system testing and analysis and enable a strong form of loose coupling (i.e., minimizing interdependencies among modules).

Modularity and layering are effective in managing the complexity of the composed system. They provide the means to decompose the system into discrete and aggregate elements to better comprehend the system in terms of its structure, flows, relationships, and how the system delivers the required capability. The structured composition of the constituent elements must also adhere to the principle of *Compositional Trustworthiness* to provide a basis to support claims about how the system is composed based on the application of modularity, layering, and partially ordered dependencies to achieve authorized and intended behaviors and outcomes.

References: [2] [20] [46] [48] [49]

E.29. Substantiated Trustworthiness

Principle: System trustworthiness judgments are based on evidence that the criteria for trustworthiness have been satisfied.

Note: Trustworthiness should not be assumed but rather substantiated through evidence that clearly shows the extent to which an entity is worth being trusted. This helps to ensure that an entity is never trusted beyond the extent to which it is worthy of trust. The approach to substantiated trustworthiness requires *Commensurate Rigor* with cautious mistrust (i.e., system elements are assumed to be guilty until proven innocent).[84]

Substantiated trustworthiness is characterized by a design mentality in which all components involved in the design context (i.e., a system element and the elements with which it interacts) are treated with a mutually suspicious mindset [2] [20]. Such mutual suspicion reflects cautious mistrust – the feeling or thought that something undesired, unwanted, or unexpected is possible or can happen. The design for every system element should reflect a lack of trust in interacting elements or itself. This suspicion assumes element non-performance and addresses the following cases:

- *Interacting element suspicion (mutual suspicion):* The system element-of-interest design is based on the non-performance of the elements it interacts with and how their non-performance can influence the element-of-interest's behavior and outcomes. Designing to mutual suspicion is reinforced by applying the principle of *Least Privilege* to all entities (so that an element executes with only the privileges needed, mitigating harm that may be created) while applying the principle of *Least Persistence* so that each element is minimally exposed.
- *Self-suspicion:* The design for the system element-of-interest must consider its own non-performance independent of any external influence. Designing to self-suspicion may involve self-monitoring and built-in actions, including built-in testing at the initiation of the element.

This approach forces the system designer to assume that things will not go right and to rigorously seek evidence that demonstrates the effectiveness of the design when things go wrong. Considerations for system element non-performance include:

- An expectation that elements will behave and produce outcomes that are inconsistent with their design intent
- The constraints, assumptions, and preconditions that are associated with achieving threshold performance
- Intentional and unintentional events and conditions, typically referred to by terms like fault, error, failure, and compromise

References: [2] [21] [50]

E.30. Trustworthy System Control

Principle: The design for system control functions conforms to the properties of the generalized reference monitor.

Note: The trustworthy system control principle reflects the generalization of the reference monitor concept to provide a uniform design assurance basis for trustworthy system control mechanisms or constraint-enforcing mechanisms that compose to provide system control functions.

[84] Adapted from a statement made by John Rushby, SRI International, about the need for software to be treated as "guilty until proven innocent" at a Layered Assurance Workshop (LAW).

The reference monitor concept (Section D.4.2) is a foundational access control concept for secure system design. It is defined as a trustworthy abstract machine that mediates all accesses to objects by subjects [38]. As a concept for an abstract machine, the reference monitor does not address any specific implementation. A reference validation mechanism, which is a combination of hardware and software, realizes the reference monitor concept to provide the access mediation foundation for a secure system [37].

The reference monitor concept has several criteria that provide design assurance of its realization as a reference validation mechanism:

- The reference validation mechanism must be tamper-proof to ensure that its integrity and validity are not destroyed.
- The reference validation mechanism must always be invoked, and if it cannot be, then the group of programs for which it provides validation services must be considered part of the reference validation mechanism and be subject to the first and third requirements.
- The reference validation mechanism must be subject to rigorous analysis and tests, the completeness of which can be assured (with the purpose of ascertaining that the reference validation mechanism works correctly in all cases).

For trustworthy system control, a fourth criterion of non-bypassability is added (Section D.4.2).

Successful achievement of these criteria will prevent the interference of outside entities on a protection mechanism or controller. Specifically:

- A protection mechanism or feature should not be circumventable (i.e., the mechanism should be non-bypassable).
- A protection mechanism or feature should be evaluatable (i.e., sufficiently small and simple enough to be assessed to produce adequate confidence in the protection provided, the constraint or control objective enforced, and the correct implementation of the mechanism [*Reduced Complexity*]).
- A protection mechanism or feature is always invoked, providing continuous protection.
- A protection mechanism or feature must be tamper-proof (i.e., neither the protection functions nor the data that the functions depend on can be modified without authorization).

Trustworthy system control also encompasses control, safety, and security concepts to establish a system capability that sufficiently:

- Enforces constraints to achieve only the authorized and intended system behaviors and outcomes
- Provides self-protection against targeted attacks on the system
- Is absent of self-induced emergent, erroneous, unsafe, and non-secure control actions

Such a system capability underlies the loss control objectives and transforms the approach for design to not rely on having detailed knowledge of the capability, means, and methods of an adversary. This design approach can be employed in attack-dependent or attack-independent manners based on the limits of certainty for what is known with confidence about the adversary.

Trustworthy system control serves well as the design basis for individual system elements, collections of elements, networks, and systems where intentional and unintentional adversity can prevent the achievement of the loss control objectives. The principle also drives the need for rigor in engineering activities commensurate to the trust placed in the system elements.

References: [21] [35] [37] [38]

Appendix F. Trustworthiness and Assurance

The trustworthiness of a system is based on the concept of assurance. Assurance is the grounds for justified confidence that a claim or set of claims has been or will be achieved [61]. Justified confidence is derived from objective evidence that reduces uncertainty to an acceptable level and, in doing so, reduces the associated risk (Section F.2).[85] Evidence is produced by engineering verification and validation methods.[86] The evidence must be relevant, accurate, credible, and of sufficient quantity to enable reasoned conclusions and consensus among subject-matter experts that the claims are satisfied. The relationship between evidence and claims can be represented in many ways. Section F.2 discusses these approaches.

> "The trust we place in our digital infrastructure should be proportional to how trustworthy and transparent that infrastructure is and to the consequences we will incur if that trust is misplaced."
>
> **-- Executive Order (EO) on Improving the Nation's Cybersecurity [1]**
> May 2021

F.1. Trust and Trustworthiness

As discussed in Section 2.3, trust and trustworthiness are foundational concepts to engineering trustworthy secure systems, to the decisions made to grant trust, and to the extent to which trust is granted based on demonstrated trustworthiness. Trust is a belief that an entity meets certain expectations and can, therefore, be relied upon. A trustworthy entity requires sufficient evidence to support its trustworthiness claims. Trustworthiness is demonstrated based on evidence that supports a stated claim or judgment of being worthy to be trusted [2] [20] [21].

Trust in an entity can occur without a basis for or knowledge of the entity's trustworthiness. This may occur because (1) there is no alternative (e.g., an individual trusts the components involved in an Internet transaction without knowing anything about the components), (2) the need for trustworthiness is not realized and occurs de facto, or (3) other reasons (e.g., miscommunication or misrepresentation of evidence) [58]. Since the decision to trust an entity is not necessarily based on a judgment of trustworthiness, the decision to trust an entity should consider the significance (i.e., consequences, effects, and impacts) of trust expectations not being fulfilled. The criteria to grant trust are used to determine the trustworthiness of an entity. Trust granted without establishing the required trustworthiness is a significant contributor to risk.

F.1.1. Roles of Requirements in Trustworthiness

Trustworthiness judgments are based on expectations to be fulfilled by the entity to be trusted. The expectations of trustworthiness of the system, inclusive of its elements, are found in the system capability, performance, security, and other requirements. These judgments are

[85] This includes risks attributed to poor, incorrect, and unjustified decisions.
[86] These methods include combinations of demonstration, inspection, analysis, and testing.

meaningful only to the extent to which the trustworthiness-relevant requirements accurately reflect the problem, accurately define the solution, and can be verified as being satisfied by the solution.

The trustworthiness requirements about security derive from the protection needs, priorities, constraints, and concerns associated with the system's ability to achieve authorized and intended behaviors and outcomes, deal with adversity, and control loss. The requirements also address the measures used to assess trustworthiness and the evidentiary data and information required to substantiate trustworthiness conclusions and grant trust. The requirements engineering discipline provides the methods, processes, techniques, and tools for this to occur.

> "A meaningful claim of trustworthiness cannot be based on an isolated demonstration that the system contains a protection capability assumed to be effective or sufficient. Instead, conclusions about a protection capability must have their basis on evidence that the system was properly specified, designed, and implemented with the rigor needed to deliver a system-level function in a manner deemed to be trustworthy and secure." [2]

F.1.2. Design Considerations

The design for a trustworthy secure system requires the application of principled engineering concepts and methods supported by evidence that provides assurance that all security-relevant claims about the system are satisfied (Section F.2).[87] Some considerations that apply to achieving trustworthiness in system design are:

- **Composition**

  Trustworthiness judgments are compositional. They must align with how the set of composed elements provides a system capability. The way that the system is composed from its system elements must include the design principles of *Compositional Trustworthiness* and, to the extent practical, *Structured Decomposition and Composition*.

- **States, Modes, and Transitions**

  Ideally, the implemented system design will result in a system that continually remains in secure states and modes with secure transitions between states and modes (Section 3.2). Realistically, the system will have insecure and indeterminant (i.e., unknown if secure or insecure) systems states and modes. The design must account for these cases and provide the capability to transition from insecure and indeterminant states and modes to secure states and modes (*Protective Recovery*).

[87] Constraints and claims are expressed in terms of functional correctness, strength of function, concerns for asset loss and consequences, and the protection capability derived from adherence to standards or from the use of specific processes, procedures, or methods.

- **Failure Propagation**

  All systems fail at some point. When a failure occurs, another failure scenario or the creation of a new failure scenario should not be triggered or invoked (*Protective Failure*). Designing without single points of failure (*Redundancy*) – including not having common mode failures (*Diversity*) – can help isolate system element failures while providing the required system capabilities. Additionally, the response to failure should not lead to loss or other failures (*Protective Recovery*).

- **Anomaly Detection**

  Anomaly Detection provides situational awareness that allows the system to decide and recommend corrective actions to account for actual and potential deviations from accepted norms.

- **Trades**

  Not every system element has trustworthiness that is sufficient for its intended purpose. A deficiency in trustworthiness can result from:

  - Technical feasibility and practicality issues
  - Cost and schedule issues of what is feasible and practical
  - The limits of certainty (i.e., what is not known, what cannot be known, and what is underappreciated [known or could be known but dismissed prematurely])

  The trade space is the rigorous application of the design principles that provide a basis for the necessary design decisions to maximize the trustworthiness of individual system elements and aggregates of elements. For example, in addressing the feasibility and practicality of cost and schedule issues, the design principle of minimizing the number of system elements that must be trusted (*Minimal Trusted Elements*) is applied. This reduces the size and scope of the effort and potentially reduces the expense of generating evidence of trustworthiness.

F.2. Assurance

Assurance is the grounds for justified confidence that a claim or set of claims has been or will be achieved [61]. Assurance is a complex and multi-dimensional property of the system that builds over time. Assurance must be planned, established, and maintained in alignment with the system throughout the system life cycle.

Adequate security judgments should be based on the level of confidence in the ability of the system to protect itself against asset loss and the associated consequences across all forms of adversity.[88] It cannot be based solely on individual efforts, such as demonstrating compliance, functional testing, or adversarial penetration tests. Judgments include what the system cannot do, will not do, or cannot be forced to do. These judgments of non-behavior must be grounded in sufficient confidence in the system's ability to correctly deliver its intended function in the presence and absence of adversity and to do so when used in accordance with its design intent.

[88] The term *adversity* refers to those conditions that can cause a loss of assets (e.g., threats, attacks, vulnerabilities, hazards, disruptions, and exposures).

The needed evidentiary basis for such judgments derives from well-formed and comprehensive evidence-producing activities that address the requirements, design, properties, capabilities, vulnerabilities, and effectiveness of security functions. These activities include a combination of demonstration, inspection, analysis, testing, and other methods required to produce the needed evidence. The evidence acquired from these activities informs reasoning by qualified subject-matter experts to interpret the evidence to substantiate the assurance claims made while considering other emergent properties that the system may possess.

VENEER SECURITY

Assurance is difficult but necessary.

"I've covered a lot of material in this book, some of it quite tricky. But I've left the hardest parts to the last. First, there's the question of assurance …" [5].

Veneer security is security functionality provided without corresponding assurance so that the functionality only appears to protect resources when it does not. Veneer security results in a false sense of security and, in fact, increases risk due to the uncertainty about the behavior and outcomes produced by the security functionality in the presence and absence of adversity. Veneer security must be avoided [62].

Compliance is a form of "veneer security." While compliance may have an important informing role in judgments of trustworthiness, compliance-based judgments – like other forms of veneer security – do not suffice as the sole evidentiary basis for assurance and the associated judgments of trustworthiness.

F.2.1. Security Assurance Claims

From a security perspective, a top-level claim addresses freedom from the conditions that cause asset loss and the associated consequences. Specifically, this means the system will adequately contribute to freedom from the conditions that cause asset loss and the associated consequences.

Top-level claims decompose in a structured manner into subclaims about the desired attributes of a trustworthy secure system. Subclaims address the requirements, design, implementation, methods, and adversities that demonstrate that the system adequately contributes to ensuring only authorized and intended system behaviors and outcomes. These subclaims are derived from concerns about the completeness and accuracy of stakeholder and system requirements,[89] enforcement of the security policy, proper implementation of the design, proper maintenance of the system, the usability of the system,[90] and the avoidance, minimization, and mitigation of

[89] Claims are not expressed solely as a restatement of the security functional and performance requirements. Doing so only provides assurance that the security requirements are satisfied with the implicit assumption that the requirements are correct, provide adequate coverage, and accurately reflect stakeholder needs and concerns.

[90] Most system failures have a human component. Thus, assurance must consider human frailty [5]. Operator behavior is a product of the environment (including its systems) in which it occurs [36].

defects, errors, and vulnerabilities.[91] Other subclaims may exist involving the ability to exhibit predictable behavior while operating in secure states in the presence and absence of adversity and the ability to recover from an insecure state. Claims can be expressed in terms of functional correctness, strength of function, and the protection capability derived from adherence to standards and/or from the use of specific processes, procedures, and methods.

LEARNING FROM SAFETY

The NASA System Safety Handbook [6] describes the relevant claims to be met in terms of the top-level claim that the system is adequately safe with subclaims, including that the system is designed to be as safe as reasonably practicable, built to be as safe as reasonably practicable, and operated as safely as reasonably practicable.

F.2.2. Approaches to Assurance

There are three general approaches to assurance. These assurance approaches can vary based on the type of evidence, how the evidence is acquired, the strength of the judgments made based on the acquired evidence, and the extent to which the assurance matches decision-making needs. From weakest to strongest, the assurance approaches are axiomatic, analytic, and synthetic.

- **Axiomatic Assurance** (assurance by assertion) is based on beliefs accepted on faith in an artifact or process. The beliefs are often accepted because they are not contradicted by experiment or demonstration. Axiomatic assurance is not suited to complex scenarios [62].
  - Demonstration of conformance and compliance are types of axiomatic assurance. While useful, they are not well-suited as the sole basis of assurance for complex scenarios.
- **Analytic Assurance** (assurance by test and analysis) derives from testing or reasoning to justify conclusions about properties of interest. Belief is relocated from an artifact or process to trust in some method of analysis. The feasibility of establishing an analytic basis depends on the amount of work involved in performing the analysis and on the soundness of any assumptions underlying that analysis. Analytic methods are most relevant in a model that spans all relevant uses and all interfaces to the environment. That is, the model must not ignore too many details.
  - Testing demonstrates the presence but not the absence of errors and vulnerabilities. Testing and analyses will have uncertainty that cannot be ignored, especially when they lack comprehensiveness. Uncertainty contributes to risk.
- **Synthetic Assurance** (assurance by structured reasoning) derives from the method of composition of the "components of assurance" (i.e., the assurance derives from the manner of

[91] Not all vulnerabilities can be mitigated to an acceptable level. There are three classes of vulnerabilities in systems: (1) vulnerabilities whose existence is known and either eliminated or made to be inconsequential, (2) vulnerabilities whose existence is known but that are not sufficiently mitigated, and (3) unknown vulnerabilities that constitute an element of uncertainty. That is, the fact that the vulnerability has not been identified should not give increased confidence that the vulnerability does not exist. Determining the effect of vulnerabilities that are in the delivered system and the risk posed by those vulnerabilities and accepting uncertainty about the existence of a vulnerability that will only become known over time are important aspects that are addressed by assurance.

synthesis of the constituent parts). It requires that assurance be a consideration at every step of design and implementation, from the smallest components to the final subsystem realization.

- The assurance case described in [30] is an example of structured reasoning (Section 4.3). Structured reasoning serves to fill the gaps associated with the axiomatic and analytic assurance approaches. Since synthetic assurance is based on the expert judgment of available evidence, it is not complete. However, synthetic assurance does further reduce uncertainty and, thus, reduces risk.

Assurance depends on the quality of the evidence used in arguments demonstrating that claims about the system are satisfied. Assurance evidence can be obtained either directly through measurement, testing, observation, or inspection or indirectly through analysis, including the analysis of data obtained from measurement, testing, observation, or inspection. Evidence must have sufficient quality in accuracy, credibility, relevance, rigor, and quantity. The accuracy, credibility, and relevance of evidence should be confirmed prior to its use. For example, some evidence can support arguments for strength of function, others for negative requirements (i.e., what will not happen), and still other evidence for qualitative properties.

ASSURANCE CASE

An assurance case is a reasoned, auditable artifact that is created to support the contention that a top-level claim is satisfied. The assurance case includes systematic argumentation, evidence, and explicit assumptions that support the claim.

An assurance case contains the following elements [30]:

- One or more claims about properties
- Arguments that logically link the evidence and any assumptions
- A body of evidence
- Justification of the choice of a top-level claim and the method of reasoning

Assurance cases have numerous advantages over other means for obtaining confidence, such as in the areas of comprehension, informing needed allocation responsibilities, information organization, and robust due diligence [63]. These advantages were greater in areas with otherwise insufficient methods for achieving high assurance. Additionally, assurance cases were determined to be more efficient for complex and novel systems, as well as systems in need of high assurance.

Many formalizations and tools for building assurance cases have been developed in recent years, including the Goal Structuring Notation (GSN) [64] and NASA's AdvoCATE: Assurance Case Automation Toolset [65].

F.2.3. Assurance Needs

Assurance is a need that is to be engineered and satisfied similar to the need to engineer the system capability to satisfy specified capability needs. Assurance needs for trustworthy secure systems are grounded in the concerns of loss and adverse effects due to intentional and unintentional adversity (*Commensurate Trustworthiness*, *Substantiated Trustworthiness*, *Commensurate Rigor*). Assurance needs include the evidence-basis for reasoning, the degree of rigor to acquire and interpret the evidence, and the selection of the methods, tools, and processes used throughout the system life cycle. Similar to capability and performance needs, assurance needs, expectations, priorities, and constraints should be expressed as system requirements and achieved, tracked, and maintained within the systems engineering effort.

CONFIDENCE MAY BE NEGATIVE

Assurance evidence can support a conclusion that a stated claim is not achieved or that there is an insufficient basis to conclude that the claim is supported or not supported. In either case, the assurance is negative relative to the goal of substantiating the claim. That is, the system or some part of the system is not sufficiently trustworthy and should not be trusted relative to its specified function without further action.

Assurance needs determine the type of evidence and the rigor associated with the activities, methods, and tools used to acquire the evidence to satisfy the following cases:

- *What is to be accomplished in the systems engineering effort:* The realization of the design for a secure system
- *The means to conduct the systems engineering effort:* The methods, processes, and tools employed (driven by rigor and assurance objectives) to realize the design for a secure system
- *The results of the systems engineering effort:* The substantiated effectiveness of the realized design of the secure system

Assurance needs can vary and constitute a trade space that must be managed similar to how capability and performance needs can vary. The degree of rigor is the primary means of varying assurance. As shown in Figure 17, a direct relationship exists between the degree of rigor and assurance and the stakeholder's assessment of the effects of asset loss. The assurance trade space includes the following considerations:

- Cost, schedule, and performance
- Architecture and design decisions
- Selection of technology and solutions
- Selection and employment of methods and tools
- Qualifications necessary for subject-matter experts

Requirements analysis across stakeholder and system requirements determines the threshold degree of rigor that is required. When a system cannot practicably meet the needed degrees of rigor, stakeholders should have a means to determine if they will accept the associated risk.

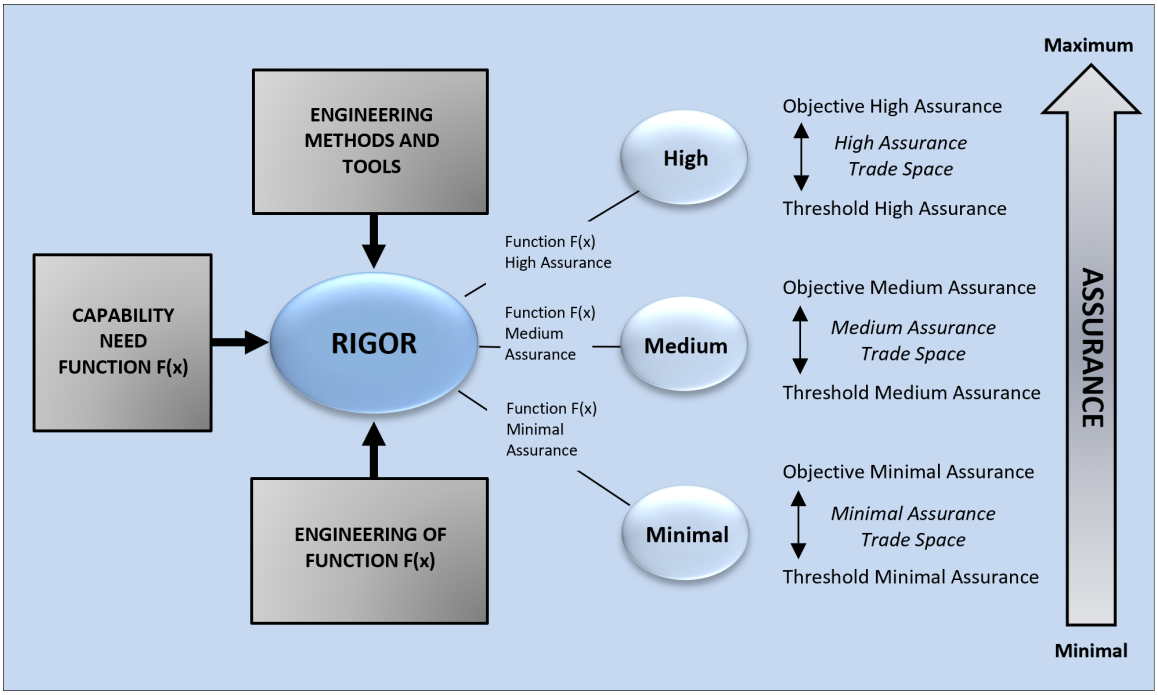

Fig. 17. Assurance and Degree of Rigor in Realizing a Capability Need

The highest levels of rigor across systems can require formal methods – techniques that model systems as mathematical entities to enable rigorous verification of the system's properties through mathematical proofs. Formal methods depend on formal specifications (i.e., statements in a language whose vocabulary, syntax, and semantics are formally defined) and a variety of models, including a formal security policy model (i.e., a mathematically rigorous specification of a system's security policy [Appendix C]).

Due to associated costs and complexity, formal methods are typically limited to engineering efforts where only the highest levels of assurance are needed, such as the formal modeling, specification, and verification of security policy and the implementation that enforces the policy (Section D.4.2). In this case, the security policy model is verified as complete for its scope of control and as self-consistent. The verified security policy model then serves as a foundation to verify the models of the design and implementation of the mechanisms that provide for decision-making and the enforcement of those decisions.

Appendix G. System Life Cycle Processes Overview

This appendix provides an overview of the system life cycle processes in [4], establishes the basis for the in-depth coverage of those processes in subsequent appendices, and describes relevant relationships among the various process groups and processes (Section G.2).

G.1. Process Overview

The activities performed during the system life cycle are grouped into Technical Processes (Appendix H), Technical Management Processes (Appendix I), Organizational Project-Enabling Processes (Appendix J), and Agreement Processes (Appendix K) [4]. Appendices H, I, J, and K describe the considerations and contributions to the system life cycle processes to achieve trustworthy secure systems. Table 5 lists the four process groups and processes in each group [4].

Table 5. System Life Cycle Processes

TECHNICAL PROCESSES	TECHNICAL MANAGEMENT PROCESSES	ORGANIZATIONAL PROJECT-ENABLING PROCESSES	AGREEMENT PROCESSES
- Business or Mission Analysis (BA) - Stakeholder Needs and Requirements Definition (SN) - System Requirements Definition (SR) - System Architecture Definition (SA) - Design Definition (DE) - System Analysis (SA) - Implementation (IP) - Integration (IN) - Verification (VE) - Transition (TR) - Validation (VA) - Operation (OP) - Maintenance (MA) - Disposal (DS)	- Project Planning (PL) - Project Assessment and Control (PA) - Decision Management (DM) - Risk Management (RM) - Configuration Management (CM) - Information Management (IM) - Measurement (MS) - Quality Assurance (QA)	- Life Cycle Model Management (LC) - Infrastructure Management (IM) - Portfolio Management (PM) - Human Resource Management (HR) - Quality Management (QM) - Knowledge Management (KM)	- Acquisition (AQ) - Supply (SP)

The security-relevant considerations and contributions to the system life cycle are provided as systems security engineering tasks. The tasks are aligned with the engineering viewpoints of the life cycle processes and are based on the foundational security and trust principles and concepts described in Chapter Two, Chapter Three, Chapter Four, Appendix C, Appendix D, Appendix E, and Appendix F. The tasks use and leverage the principles, concepts, terms, and practices of systems engineering to help facilitate consistency in their application as part of a systems engineering effort.

The system life cycles processes, activities, and tasks are to be applied as needed. They are not dependent on, oriented to, or presumed to be used or needed in any specific system development methodology. By design, the processes and their activities and tasks can be applied concurrently,

iteratively, or recursively at any level in the structural hierarchy of a system with the appropriate fidelity and rigor and at any stage in the system life cycle in accordance with acquisition, systems engineering, or other process models. Using their expertise and experience, practitioners can tailor the system life cycle processes, activities, and tasks to achieve optimized and efficient results.[92] Considerations include:

- How the system life cycle processes apply within the development models used by an organization
- The ordering or sequencing of the activities and tasks in the system life cycle processes
- How the outcomes may be achieved in ways that do not strictly adhere to the presentation of the processes in this publication
- Additional activities and tasks needed to achieve specific outcomes
- The size, scope, and complexity of the system
- The need to accommodate specific technologies, methods, or techniques used to develop the system

Tailoring the system life cycle processes allows the engineering team to:

- Optimize the application of the processes in response to technological, programmatic, acquisition, process, procedural, system life cycle stage, or other objectives and constraints
- Allow for the concurrent application of the processes by sub-teams focused on different parts of the same engineering effort
- Facilitate the application of the processes to conform with a variety of system development methodologies, processes, and models (e.g., agile, spiral, waterfall) that could be used on a single engineering effort
- Accommodate the need for unanticipated or other event-driven execution of the processes to resolve issues and respond to changes that occur during the engineering effort

While the life cycle processes and activities are restated from [4], the tasks in this publication are neither a restatement of nor a one-for-one mapping to the tasks in [4]. This publication focuses on the security contributions to the processes, and, therefore, the tasks are titled to reflect these contributions. In some cases, tasks have been added to address the range of outcomes appropriate for achieving trustworthy secure system objectives.

The descriptions of the system life cycle processes assume that sufficient time, funding, and human and material resources are available to ensure the complete application of the processes within the systems engineering effort. The processes represent the *standard of excellence* within which appropriate tailoring is accomplished to achieve realistic, optimal, and cost-effective results within the constraints imposed on the engineering team.

[92] Tailoring can occur as part of the project planning process at the start of the systems engineering effort or in an ad hoc manner at any time during the engineering effort when situations and circumstances so dictate. Understanding the fundamentals of systems security engineering (i.e., the science underpinning the discipline) helps to inform the tailoring process whenever it occurs during the system life cycle. The INCOSE Systems Engineering Handbook provides additional guidance on how to tailor the systems engineering processes [15].

Each of the system life cycle processes contains a set of activities and tasks that produce security-focused outcomes.[93] These outcomes combine to deliver a system and corresponding body of evidence[94] that serve as the basis to:

- Substantiate the security and trustworthiness of the system
- Identify and assess security-relevant risk across stakeholder concerns with respect to the use of the system in support of mission or business objectives
- Provide inputs to other processes associated with delivering the system
- Determine operations and sustainment strategies and actions to address the risk delivered with the system

Each system life cycle process description has the following sections:

- *Life Cycle Purpose:* Describes the objective of performing the process
- *Security Purpose:* Establishes what the process achieves from the security perspective
- *Security Outcomes:* Expresses the security-relevant observable results expected from the successful performance of the process and the data generated by the process
- *Security Activities and Tasks:* Provides a set of security-relevant activities and tasks that support achieving security outcomes for the process[95]

The outcomes described are achieved by personnel, processes, and technology. Personnel conduct activities and tasks throughout the stages of the system life cycle to produce outcomes that achieve the defined security objectives. No single personnel role is responsible for producing all outcomes stated in the system life cycle processes (i.e., the processes are not role-specific). Thus, multiple roles may contribute to a specific outcome.

Finally, this publication describes systems engineering considerations to produce the specified outcomes. However, no specific roles or responsibilities are identified. Organizations define and allocate roles and responsibilities to the personnel who execute the life cycle processes. Figure 18 provides an example of personnel categories, each with a scope of authority, control, roles, and responsibilities that collectively achieve the outcomes for the category. The outcomes produced across all personnel categories achieve the defined security objectives.

[93] Outcomes inform other processes, including those external to the engineering effort (e.g., the organizational life cycle processes of stakeholders and certification, authorization, or regulatory processes).

[94] The comprehensiveness, depth, fidelity, credibility, and relevance of the body of evidence are factors in achieving the desired level of assurance. The objective is a body of evidence sufficient to convince stakeholders that their assurance needs are satisfied.

[95] The activities and tasks are accomplished cooperatively within and across various roles of the organization, inclusive of systems security engineering. While this publication focuses on systems security engineering, it does not fulfill all aspects of every activity and task.

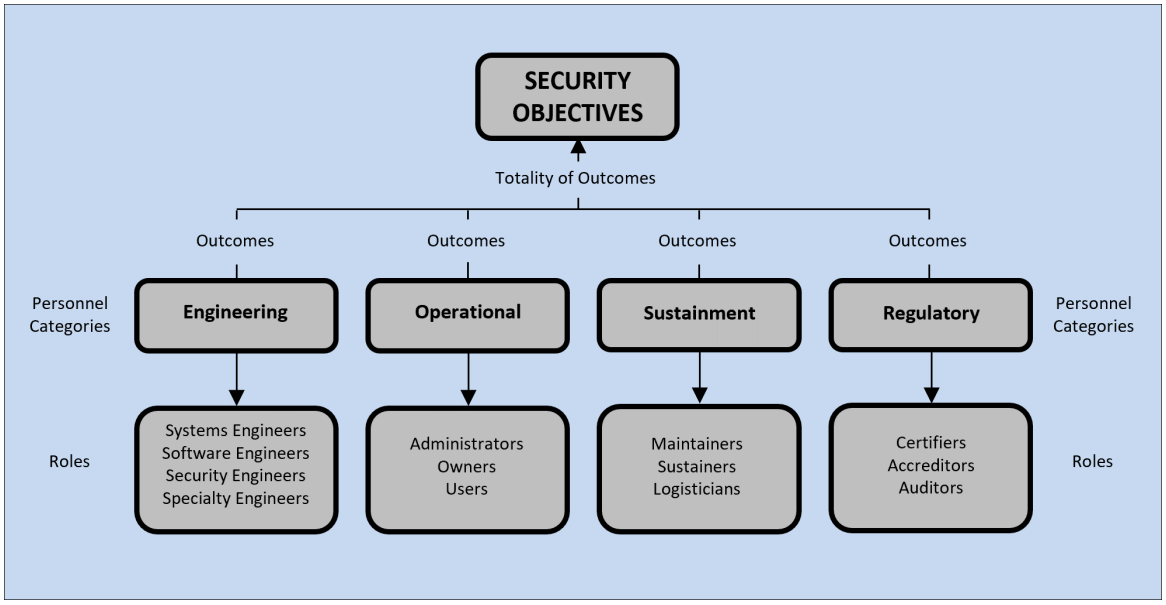

Fig. 18. Types of Personnel and Roles that Support Life Cycle Processes

G.2. Process Relationships

Figure 19 illustrates common logical relationships among system life cycle process groups and processes that can be used as a framework and altered as necessary as part of tailoring [4].

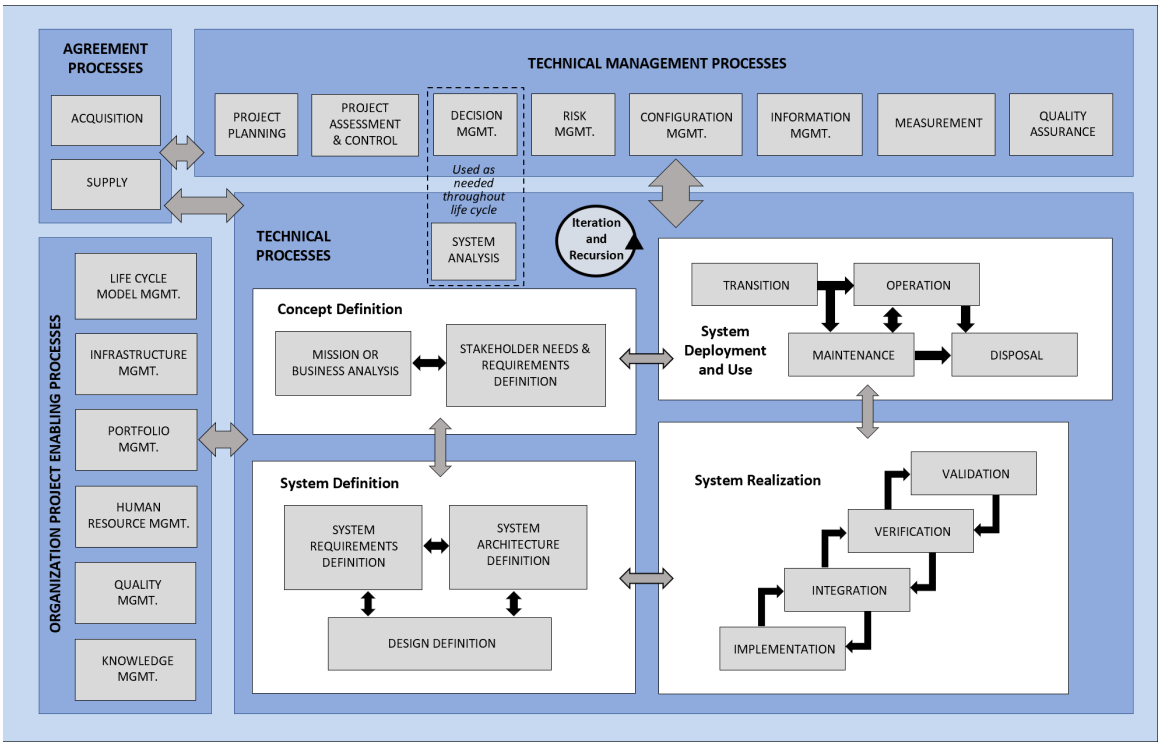

Fig. 19. Relationships Among Life Cycle Processes

Process relationships are further illustrated by the *use cases* in [94]. Several prominent use cases include:

- *Establish a formal agreement*
  - Agreements between organizations, between projects, and for work efforts within a project
  - Commonly a formal contract between an acquirer and the supplier, including a prime contractor and its subcontractors
- *Satisfy an agreement*
  - Processes to satisfy the agreement, including information that a supplying organization provides to the acquiring organization to ensure compliance with the agreement
- *Engineer a system of interest*[96]
  - Relationships among the technical processes (Appendix H)[97]

The following sources provide additional information on system life cycle processes and their relationships: [4] [13] [15] [17] [18] [19] [27] [94] [95] [96]. Processes for software-intensive systems are discussed in [86].

[96] The application of technical processes for engineering a system of interest will occur recursively to realize subsystems and system elements. See Annex A of [96] for additional details.

[97] This use case often supports satisfying an agreement.

Appendix H. Technical Processes

This appendix contains the *Technical Processes* from [4] with security-relevant considerations and contributions for the purpose, outcomes, activities, and tasks. As noted in Section G.2, the application of these processes at any life cycle stage is described in [96], which has a set of example stages and stage outcomes for enacting technical processes within system and software life cycles.

H.1. Business or Mission Analysis

The purpose of the *Business or Mission Analysis* process is to define the overall strategic problem or opportunity, characterize the solution space, and determine potential solution class(es) that can address a problem or take advantage of an opportunity.

H.1.1. Security Purpose

- Define the security aspects related to the strategic problems or opportunities.
- Identify the security objectives, concerns, and constraints that inform the potential solution class(es).

H.1.2. Security Outcomes

- Security aspects of the strategic problem or opportunity space are defined.
- Security aspects of the solution space are characterized.
- The definition of the preliminary operational concepts and other concepts in the life cycle stages is informed by the security aspects of the problem or opportunity space.
- Alternative solution class(es) that consider(s) identified security aspects is/are analyzed.
- Selection of the preferred alternative solution class(es) is informed by the security aspects of the solution space.
- Enabling systems or services needed for the security aspects of business or mission analysis are available.
- Traceability of the security aspects of the strategic problems and opportunities to the preferred alternative solution class(es) is established.

H.1.3. Security Activities and Tasks

BA-1 PREPARE FOR BUSINESS OR MISSION ANALYSIS

BA-1.1 Identify the security aspects for enabling systems or services needed to support business or mission analysis.

BA-1.2 Identify and plan for enabling systems or services needed to support the security aspects of business or mission analysis.

BA-1.3 Obtain or acquire access to the security aspects of enabling systems or services to be used in business or mission analysis.

References: [4] [97]

BA-2 DEFINE THE PROBLEM OR OPPORTUNITY SPACE

BA-2.1 Analyze the problems or opportunities in the context of the security-relevant trade space factors.

Note: The security-relevant trade space factors are analyzed within the context of all factors, including factors related to loss tolerances. The results of the analyses inform decisions on the suitability and feasibility of alternative options to be pursued.

BA-2.2 Define the security aspects of the mission, business, or operational problem or opportunity to be addressed by the solution class(es).

Note: Information is elicited from stakeholders to acquire an understanding of the mission, business, or operational problem or opportunity from a system security perspective. Security aspects include security objectives, concerns, and constraints.

References: [4] [30] [61] [97] [98] [99]

BA-3 CHARACTERIZE THE SOLUTION SPACE

BA-3.1 Define the security aspects of the preliminary operational concepts and other concepts in life cycle stages.

Note 1: Security operational concepts include modes of secure operation, security-relevant operational scenarios and use cases, and secure usage within a mission area or line of business.

Note 2: Security aspects are integrated into the life cycle concepts and used to support feasibility analysis and the evaluation of candidate alternative solution class(es).

BA-3.2 Identify the security aspects of the alternative solution classes.

References: [4] [71] [96] [97]

BA-4 EVALUATE ALTERNATIVE SOLUTION CLASSES

BA-4.1 Assess each alternative solution class while considering the identified security aspects.

BA-4.2 Select the preferred alternative solution class (or classes) based on the identified security aspects, trade space factors, and other criteria defined by the organization.

BA-4.3 Provide security-relevant feedback to strategic-level life cycle concepts to reflect the selected solution class(es).

References: [4] [71] [96] [97]

BA-5 MANAGE THE BUSINESS OR MISSION ANALYSIS

BA-5.1 Maintain traceability of the security aspects of business or mission analysis.

Note: Bidirectional traceability is maintained between identified security aspects and supporting security data associated with the problems and opportunities, proposed solution class or classes, and organizational strategy.

BA-5.2 Provide the security-relevant artifacts that have been selected for baselines.

References: [4] [71] [96]

H.2. Stakeholder Needs and Requirements Definition

The purpose of the *Stakeholder Needs and Requirements Definition* process is to define the stakeholder requirements for a system that can provide the capabilities needed by users and other stakeholders in a defined environment.

H.2.1. Security Purpose

- Identify the protection needs associated with the stakeholder needs and requirements for a system that can protect the capabilities needed by users and other stakeholders in a defined environment.

H.2.2. Security Outcomes

- Security-relevant stakeholders of the system are identified.
- Security concerns of stakeholders are identified.
- Required characteristics and context for the secure use of capabilities for system life cycle concepts in system life cycle stages are defined.
- Stakeholder assets and asset classes are identified.
- Adversities presented by the environment are characterized.
- Asset protection priorities are determined.
- Stakeholder protection needs are defined.
- Security-driven and security-informed constraints on a system are identified.
- Prioritized stakeholder protection needs are transformed into stakeholder requirements.
- Security-oriented performance measures and quality characteristics are defined.
- Stakeholder agreement that their protection needs and expectations are adequately reflected in the requirements is achieved.
- Enabling systems or services needed for the security aspects of stakeholder needs and requirements definition are available.
- Traceability of stakeholder requirements to stakeholders and their protection needs is established.

H.2.3. Security Activities and Tasks

SN-1 PREPARE FOR STAKEHOLDER NEEDS AND REQUIREMENTS DEFINITION

SN-1.1 Identify the stakeholders and their security concerns.

Note 1: All stakeholders have security concerns. Some concerns are explicitly known and can be stated; others are initially implicit, not necessarily known, and must be made explicit through discourse.

Note 2: This includes stakeholders who represent milestone decision authority, regulatory, certification, authorization, acceptance, and similar organizations with specific security-relevant decision-making authority and responsibilities.

SN-1.2 Define the stakeholder protection needs and requirements definition strategy.

Note: The strategy includes addressing how consensus about protection needs and requirements is to be achieved among stakeholders with opposing interests.

SN-1.3 Identify the security aspects for enabling systems or services needed to support stakeholder needs and requirements definition.

SN-1.4 Identify and plan for enabling systems or services needed to support the security aspects of stakeholder needs and requirements definition.

SN-1.5 Obtain or acquire access to the security aspects of enabling systems or services to be used in stakeholder needs and requirements definition.

References: [4] [30] [61] [86] [97] [98] [99] [100]

SN-2 DEVELOP THE OPERATIONAL AND OTHER LIFE CYCLE CONCEPTS

SN-2.1 Define a representative set of scenarios to identify required protection capabilities and security measures that correspond to anticipated operational and other life cycle concepts.

Note: The scenarios reflect how the system is intended to behave in the intended operational environments. Scenarios also help to identify security-driven changes to life cycle concepts.

SN-2.2 Characterize the security aspects of the operational environments and the intended users.

Note 1: This includes distinguishing what is and is not known about adversity within the operational environments.

Note 2: This includes the trust expectations for users to address insider threat concerns. If a user security aspect cannot be obtained or there is uncertainty about the trust of users, it will significantly drive design and the operational procedure to complement the design.

SN-2.3 Identify the interactions among entities and the system and security-relevant factors affecting the interactions.

Note: The interactions among entities (e.g., personnel, enabling systems, and interfacing systems) and the system and the factors affecting the interactions need to be understood to inform engineering efforts. Factors influencing the interactions include the environment of the system of interest and any system of systems to which the system of interest belongs, as well as the characterization of the entities with which the system interacts.

SN-2.4 Identify the security-relevant constraints on a system solution.

References: [4] [30] [31] [61] [91] [97] [98] [99] [100] [101] [102] [103] [104]

SN-3 DEFINE STAKEHOLDER NEEDS

SN-3.1 Define the rules capturing authorized and intended interactions, behaviors, and outcomes.

Note: The life cycle concepts and their context inform the rules.

SN-3.2 Identify stakeholder assets and asset classes.

SN-3.3 Identify loss concerns for each identified asset and each asset class.

SN-3.4 Prioritize assets based on the adverse consequences of asset loss.

SN-3.5 Characterize adversities present in the environment.

Note: Environments that expose the system to potential adversities include test, operational, maintenance, and logistical environments. Adversities need to be avoided when possible and protected against otherwise.

SN-3.6 Identify uncertainty associated with each identified adversity.

SN-3.7 Identify stakeholder protection needs.

Note: Protection needs include their success criteria, such as measures of effectiveness (MOEs).

SN-3.8 Prioritize and down-select the stakeholder protection needs.

SN-3.9 Record the stakeholder protection needs and rationale.

References: [4] [30] [31] [61] [91] [98] [99] [100] [101] [103]

SN-4 TRANSFORM STAKEHOLDER NEEDS INTO STAKEHOLDER REQUIREMENTS

SN-4.1 Identify the security-relevant constraints on a system solution.

SN-4.2 Define stakeholder requirements in a manner consistent with security aspects and protection needs.

References: [4] [30] [31] [61] [86] [98] [99] [100] [105] [106] [107] [108] [109]

SN-5 ANALYZE STAKEHOLDER NEEDS AND REQUIREMENTS

SN-5.1 Analyze the set of stakeholder requirements with respect to the protection needs.

Note: The stakeholder requirements are analyzed to determine whether the protection needs are accurately and comprehensively expressed in both individual requirements and the set of requirements. Potential analysis characteristics include that the requirements are (1) necessary, complete, succinct, and implementation-free and (2) comprehensively address the protection needs.

SN-5.2 Define security-relevant performance and assurance measures that enable the assessment of technical achievement and their relative criticality.

Note: Determining the relative criticality of measures (e.g., measures of effectiveness) captures technical achievements and reflects stakeholder priorities.

SN-5.3 Provide feedback to applicable stakeholders from the analyzed requirements to validate that their protection needs and expectations have been adequately captured and expressed.

SN-5.4 Resolve stakeholder requirements issues related to protection needs.

Note: Any change to stakeholder requirements signifies a need to reassess protection needs and determine whether any subsequent changes are required.

References: [4] [30] [31] [61] [79] [86] [98] [99] [100] [110]

SN-6 MANAGE THE STAKEHOLDER NEEDS AND REQUIREMENTS DEFINITION

SN-6.1 Obtain explicit agreement that the stakeholder requirements satisfactorily address protection needs.

SN-6.2 Record asset protection data.

SN-6.3 Maintain traceability between stakeholder protection needs and stakeholder requirements.

SN-6.4 Provide the security-relevant artifacts that have been selected for baselines.

References: [4] [86] [100]

H.3. System Requirements Definition

The purpose of the *System Requirements Definition* process is to transform the stakeholder, user-oriented view of desired capabilities into a technical view of a solution that meets the operational needs of the user.

H.3.1. Security Purpose

- Provide an accurate and complete representation of stakeholder protection needs (as expressed in the stakeholder requirements) in the system requirements.

H.3.2. Security Outcomes

- Security aspects of the system description – including system interfaces, functions, and boundaries for a system solution – are defined.
- Security-relevant system requirements and security-driven design constraints are defined.
- Security performance measures are defined.
- Security aspects of the system requirements are analyzed.
- Enabling systems or services needed for the security aspects of the system requirements definition are available.
- Traceability of the security aspects of system requirements and associated security-relevant constraints to stakeholder requirements is established.

H.3.3. Security Activities and Tasks

SR-1 PREPARE FOR SYSTEM REQUIREMENTS DEFINITION

SR-1.1 Define the security aspects of the intended behavior and outcomes at the functional boundary of the system.

Note: The intended behavior and security properties to be realized at the functional boundary consider the characteristics of the capability provided or used, the characteristics of the entities that interact with the system of interest at the functional boundary, and the associated assurance needs.

SR-1.2 Define the security domains of the system and their correlation to the functional boundaries of the system.

SR-1.3 Define the security aspects of the system requirements definition strategy.

SR-1.4 Identify the security aspects for enabling systems or services needed to support system requirements definition.

SR-1.5 Identify and plan for enabling systems or services needed to support the security aspects of system requirements definition.

SR-1.6 Obtain or acquire access to the security aspects of enabling systems or services to be used in the system requirements definition.

References: [4] [30] [31] [61] [97] [98] [99] [100]

SR-2 DEFINE SYSTEM REQUIREMENTS

SR-2.1 Define each security function that the system is required to perform.

Note: Security functions are defined for all system states, modes, and conditions of system operation and use, including the associated transitions between system states and modes. Security functions include those oriented to delivery of capability and the ability of the system to execute while preserving its inherent security characteristics.

SR-2.2 Define the security aspects of each function that the system is required to perform.

Note: This includes the need for other system functions to be non-interfering (Section D.4.1).

SR-2.3 Define necessary security-driven implementation constraints.

Note: Security-driven constraints on the system are from adversity, uncertainty, and risk, considering performance objectives and assurance needs. These constraints are informed by stakeholder requirements, the system architecture definition, and solution limitations across the life cycle.

SR-2.4 Define necessary constraints on security implementation.

Note: Constraints on security implementation are to satisfy expectations for non-security capabilities and performance.

SR-2.5 Define system security requirements and rationale.

Note: System security requirements include security capability and functional requirements, security performance and effectiveness requirements, security assurance requirements, and implementation constraints (SR-2.3 and SR-2.4 outcomes expressed as requirements).

SR-2.6 Apply security metadata to the system security requirements.

Note: Metadata enables identification and traceability to support the analysis of completeness and consistency to determine security impact when requirements change.

References: [4] [30] [31] [61] [86] [97] [98] [99] [100] [105] [106] [107] [108] [109] [111] [112] [113]

SR-3 ANALYZE SYSTEM REQUIREMENTS

SR-3.1 Analyze the complete set of system requirements in consideration of security concerns.

Note: Requirements are analyzed to ensure that they fully and properly capture security protection and security-constraint considerations. Rationale is captured to support analysis conclusions and provide a basis to conclude that the analysis has the proper perspective and is fully aware of assumptions made. See Appendix C.

SR-3.2 Define security-driven performance and assurance measures that enable the assessment of technical achievement.

Note: Each security-driven performance measure (e.g., measure of performance and technical performance measure) is analyzed to help ensure that system requirements are met, and that project cost, schedule, or performance risks associated with any non-compliance are identified.

SR-3.3 Provide feedback from the analyzed system requirements to applicable stakeholders for security-relevant reviews.

SR-3.4 Resolve system requirements security issues.

References: [4] [30] [31] [61] [79] [86] [97] [98] [99] [100] [110]

SR-4 MANAGE THE SYSTEM REQUIREMENTS

SR-4.1 Obtain explicit agreement that system requirements express protection needs.

SR-4.2 Record key security-relevant system requirement decisions and the rationale.

SR-4.3 Maintain traceability of system requirements to their security-relevant aspects.

Note: The traceability of system requirements to protection needs; stakeholder requirements; architecture elements; interface definitions; analysis results; verification methods; allocated, decomposed, and derived requirements (in their system, system element, security protection, and security-driven constraint forms); risk and loss tolerance; and assurance and trustworthiness objectives is maintained.

SR-4.4 Provide the security-relevant artifacts that have been selected for baselines.

References: [4] [30] [31] [61] [97] [98] [99] [100]

H.4. System Architecture Definition

The purpose of the *System Architecture Definition* process is to generate system architecture alternatives, to select one or more alternative(s) that frame stakeholder concerns and meet system requirements, and to express this in a set of consistent views and models.

H.4.1. Security Purpose

- Generate the architectural concepts and properties of system architecture alternatives for the system protection capability that frame stakeholder protection concerns and meet system requirements.
- Express the architectural concepts and properties in a set of consistent views and models.
- Provide the security aspects used to select one or more architecture alternatives.

H.4.2. Security Outcomes

- The problem space is refined with respect to key stakeholder security concerns.
- Alignment of the architecture with applicable security policies, directives, objectives, and constraints is achieved.
- Concepts, properties, characteristics, behaviors, functions, and constraints that are significant to security-relevant architecture decisions about the system are allocated to architectural entities.
- Identified stakeholder protection concerns are addressed by the system architecture.
- Traceability of the security aspects of system architecture elements to key architecturally relevant stakeholder and system requirements is established.
- Security aspects of architecture views and models of the system are developed.
- Security aspects of system elements, their interactions, and their interfaces are defined.

H.4.3. Security Activities and Tasks

AR-1 PREPARE FOR SYSTEM ARCHITECTURE DEFINITION

AR-1.1 Define the security aspects of the system architecture definition strategy.

AR-1.2 Identify the set of existing security-relevant architectures or reference architectures that may have direct applicability and are to be used as guiding oversight.

AR-1.3 Establish the security aspects of the architecture description framework(s), viewpoints, and modeling templates to be used throughout the system architecture definition effort.

AR-1.4 Establish security-specific viewpoints and modeling templates to be used throughout the system architecture definition effort.

AR-1.5 Determine the security evaluation objectives and criteria with respect to the concerns of key stakeholders.

AR-1.6 Determine security evaluation methods, and integrate them with evaluation objectives and criteria.

AR-1.7 Collect and review security evaluation-related information.

AR-1.8 Identify the security aspects for enabling systems or services needed to support system architecture definition.

AR-1.9 Identify and plan for enabling systems or services needed to support the security aspects of system architecture definition.

AR-1.10 Obtain or acquire access to the security aspects of enabling systems or services to be used in system architecture definition.

References: [4] [30] [61] [71] [98] [99] [100] [117]

AR-2 CREATE THE SYSTEM ARCHITECTURE CANDIDATE(S)

AR-2.1 Establish the security aspects of architecture objectives and critical success criteria.

AR-2.2 Synthesize potential trustworthy secure solution(s) in the solution space.

AR-2.3 Characterize aspects of trustworthy secure solutions and the trade space.

AR-2.4 Formulate trustworthy secure candidate architecture(s).

AR-2.5 Capture trustworthy secure architecture concepts and properties.

AR-2.6 Relate the candidate architecture(s) to other architectures and relevant affected entities to help ensure the consistency of trustworthy secure architecture concepts and properties.

AR-2.7 Coordinate the secure use of the candidate architecture(s) by intended users.

AR-2.8 Develop the security aspects of the models and views of the candidate architecture(s).

Note: The following are some typical considerations:

- The definition of the system security context and security boundaries in terms of interfaces and interactions with external entities
- The identification of architectural entities and relationships between entities that address key stakeholder protection concerns and system security requirements
- The allocation of security concepts, security properties, security characteristics, secure behaviors, security functions, or security constraints to architectural entities
- The composition of views expressing how the architecture addresses stakeholder protection concerns and meets stakeholder and system security requirements
- The harmonization of the architecture models and views

AR-2.9 Coordinate secure use of the architecture by intended users.

References: [4] [30] [61] [71] [98] [99] [100] [117]

AR-3 EVALUATE THE SYSTEM ARCHITECTURE CANDIDATE(S)

AR-3.1 Analyze trustworthy secure architecture concepts and properties, and assess the value of the architecture in meeting stakeholder security protection concerns.

AR-3.2 Characterize the candidate architecture(s) based on trustworthy secure analysis results.

AR-3.3 Formulate security-relevant evaluation findings and recommendations.

AR-3.4 Capture and communicate security-relevant evaluation results.

AR-3.5 Relate the architecture to the other architectures and relevant affected entities to help ensure consistency in the trustworthy secure system architecture.

References: [4] [30] [61] [71] [98] [99] [117]

Related Publications: [100]

AR-4 MANAGE THE RESULTS OF SYSTEM ARCHITECTURE DEFINITION

AR-4.1 Obtain agreement on the security aspects of the architecture.

AR-4.2 Record key security-relevant system architecture decisions and the rationale.

AR-4.3 Maintain the traceability of the security aspects of the system architecture.

AR-4.4 Provide the security-relevant artifacts that have been selected for baselines.

AR-4.5 Provide support to organizational architecture governance and architecture management efforts.

References: [4] [30] [61] [71] [98] [99] [100] [117]

H.5. Design Definition

The purpose of the *Design Definition* process is to provide sufficient data and information about the system and its elements to realize the solution in accordance with the system requirements and architecture.

H.5.1. Security Purpose

- Provide sufficient detailed data and information about the security aspects of the system and its elements to realize a trustworthy secure solution in accordance with the system requirements and architecture.

H.5.2. Security Outcomes

- Security aspects of design alternatives for system elements are assessed.
- System requirements are allocated to address their security aspects.
- Security interfaces and security aspects of interfaces between system elements composing the system are defined.
- Security design characteristics of each system element are defined.
- Enabling systems or services for the security aspects of design definition are available.
- Traceability of security design characteristics is established.

H.5.3. Security Activities and Tasks

DE-1 PREPARE FOR DESIGN DEFINITION

DE-1.1 Establish the trustworthy secure aspects of the design definition strategy.

DE-1.2 Determine the security technologies required for each system element composing the system.

DE-1.3 Identify the security concerns associated with each technology required for each system element.

Note 1: This includes the security concerns due to known and potential vulnerability within or enabled by the supply chains involved with the acquisition of technologies.

Note 2: The concerns may have associated risks to record and track.

DE-1.4 Determine the necessary security and trustworthiness categories of system characteristics represented in the design.

DE-1.5 Define the precepts for trustworthy secure evolution of the system design.

DE-1.6 Identify the security aspects for enabling systems or services needed to support design definition.

DE-1.7 Identify and plan for enabling systems or services needed to support the security aspects of design definition.

DE-1.8 Obtain or acquire access to the security aspects of enabling systems or services to be used in design definition.

References: [4] [30] [61] [98] [99] [100]

DE-2 CREATE THE SYSTEM DESIGN

DE-2.1 Allocate security requirements to system elements.

DE-2.2 Transform security-relevant architectural entities and relationships into design elements.

DE-2.3 Transform security-relevant architectural characteristics into trustworthy secure design characteristics.

Note: The characteristics include or reflect the expected level of assurance.

DE-2.4 Define the necessary trustworthy secure design enablers.

DE-2.5 Examine trustworthy secure design alternatives.

DE-2.6 Refine or define the security aspects of interfaces between system elements and with external entities.

Note: The details of the defined interfaces are refined to include the security aspects. These include security and security-driven constraints applied to interfaces, interactions, and behavior between components and with external entities, such as interfacing systems (Section 2.1.2), peripheral devices, and humans interacting with the system.

DE-2.7 Develop the security aspects of design artifacts.

Note 1: Design artifacts include general and security-specific specifications, data sheets, databases, and documents.

Note 2: Design artifacts include configuration and procedures to ensure security mechanism behavior specified by system-level policy and enforced by configuration and procedures (Appendix C.3).

DE-2.8 Capture the security aspects of the design.

References: [4] [30] [61] [86] [98] [99] [100] [106] [107] [108] [109]

DE-3 EVALUTE THE SYSTEM DESIGN

DE-3.1 Analyze each system design alternative against criteria developed from expected trustworthy secure design properties and characteristics.

DE-3.2 Assess each system design alternative for how well it meets stakeholder protection needs and the security aspects of the system requirements.

Note: Assessment includes assessing configuration and procedures to ensure security mechanism behavior specified by system-level policy and enforced by configuration and procedures (Appendix C.3).

DE-3.3 Combine the security analyses and assessments in the overall evaluation to select a preferred design solution.

References: [4] [30] [61] [86] [98] [99] [100] [109]

DE-4 MANAGE THE RESULTS OF DESIGN DEFINITION

DE-4.1 Obtain agreement on the security aspects of the design.

DE-4.2 Map the trustworthy secure design characteristics to the system elements.

DE-4.3 Record the trustworthy secure design decisions and the rationale.

DE-4.4 Maintain traceability of the security aspects of the system design.

Note: Traceability is maintained between the trustworthy secure design characteristics and the security architectural entities, system element requirements, interface definitions, analysis results, and verification and validation methods or techniques.

DE-4.5 Provide the security-relevant artifacts that have been selected for baselines.

References: [4] [100] [106] [107] [108]

H.6. System Analysis

The purpose of the *System Analysis* process is to provide a rigorous basis of information and data for technical understanding to aid decision-making and technical assessments across the life cycle.

H.6.1. Security Purpose

- Produce a rigorous basis of data and information for the technical understanding of security aspects to aid decision-making and technical assessments across the life cycle.

H.6.2. Security Outcomes

- Security aspects of system analysis needs are identified.

- Security aspects of system analysis assumptions and results are validated.
- System analysis results provided for all decisions or technical assessment needs include security aspects.
- Enabling systems or services for the security aspects of system analysis are available.
- Traceability of the security aspects of the system analysis results is established.

H.6.3. Security Activities and Tasks

SA-1 PREPARE FOR SYSTEM ANALYSIS

SA-1.1 Define the security aspects of the system analysis strategy.

SA-1.2 Identify the security aspects of the problem or question that require system analysis.

Note: The problem or question may not be driven by or have obvious security considerations or aspects.

SA-1.3 Identify the security-relevant stakeholders of the system analysis.

SA-1.4 Define the scope, objectives, level of fidelity, level of rigor, and level of assurance for the security aspects of system analysis.

SA-1.5 Select the methods to address the security aspects of system analysis.

SA-1.6 Identify the security aspects for enabling systems or services needed to support system analysis.

SA-1.7 Identify and plan for enabling systems or services needed to support the security aspects of system analysis.

SA-1.8 Obtain or acquire access to the security aspects of enabling systems or services to be used in system analysis.

SA-1.9 Identify and validate security-relevant assumptions.

Note 1: This includes assumptions derived from the limits of certainty: what is known, what is insufficiently known, and what is unknown.

Note 2: Assumptions that cannot be validated represent uncertainty and potential risk.

SA-1.10 Plan for and collect the data and inputs needed for the security aspects of the analysis.

References: [4] [30] [61] [98] [99] [100]

SA-2 PERFORM SYSTEM ANALYSIS

SA-2.1 Apply the selected analysis methods to perform the required security-relevant aspects of system analysis.

SA-2.2 Review analysis results for security-relevant quality and validity.

Note: The results are coordinated with associated and previously completed security-relevant analyses. Trustworthiness of the results is determined with the review.

SA-2.3 Establish conclusions and recommendations for the security aspects of the system analysis.

Note: Subject-matter experts are consulted and participate in the formulation of conclusions and recommendations.

SA-2.4 Record the results of the security aspects of the system analysis.

References: [4] [86] [100] [106] [107] [108] [109]

SA-3 MANAGE SYSTEM ANALYSIS

SA-3.1 Maintain traceability of the security aspects of the system analysis results.

Note: Bidirectional traceability captures the relationship between the security aspects of the system analysis results, the methods employed, the data used for the analysis, the assumptions, and the context that defines the problem or question addressed.

SA-3.2 Provide the security-relevant artifacts that have been selected for baselines.

Note: This includes general artifacts and security-specific artifacts.

References: [4] [100] [106] [107] [108]

H.7. Implementation

The purpose of the *Implementation* process is to realize a specified system element.

H.7.1. Security Purpose

- Transform system security requirements, architecture, and design (including interfaces) into actions that create a trustworthy secure system element according to the practices of the selected implementation technology using appropriate security and non-security technical specialties or disciplines.

H.7.2. Security Outcomes

- Security-relevant implementation constraints that influence the requirements, architecture, or design are identified.
- A trustworthy secure system element is realized.
- System elements are securely packaged and stored.
- Enabling systems or services for the security aspects of implementation are available.
- Traceability of the security aspects of the implemented system elements is established.

H.7.3. Security Activities and Tasks

IP-1 PREPARE FOR IMPLEMENTATION

IP-1.1 Define the trustworthy secure aspects of the implementation strategy.

Note 1: These aspects apply to all system elements that are acquired new, built new, or reused (with or without modification). If the strategy is reuse, the project determines the extent,

source, suitability, and trustworthiness of the reused system elements for the new purpose. The implementation strategy includes procedures, fabrication processes, tools and equipment, tolerances, and verification uncertainties, which may introduce weaknesses and vulnerabilities. In the case of repeated system element implementation (e.g., mass production, replacement system elements), the procedures and fabrication processes are defined to achieve consistent and repeatable trustworthy producibility.

Note 2: The security aspects are informed by the targeted level of assurance, security verification uncertainties, and security concerns associated with implementation-related logistics, supply, and distribution of components.

IP-1.2 Identify security-relevant constraints and objectives from implementation in the system security requirements, architecture and design characteristics, or implementation techniques.

IP-1.3 Identify the security aspects for enabling systems, services, and materials needed to support implementation.

IP-1.4 Identify and plan for enabling systems, services, and materials needed to support the security aspects of implementation.

IP-1.5 Obtain or acquire access to the security aspects of enabling systems, services, and materials to be used in implementation.

References: [4] [30] [61] [98] [99] [111] [112] [113]

IP-2 PERFORM IMPLEMENTATION

IP-2.1 Realize or adapt system elements in accordance with the security aspects of the implementation strategy and implementation procedures, as well as security-relevant constraints.

Note: System elements can include:

- *Hardware and Software:* Hardware and software elements are either acquired or fabricated.

  Custom hardware fabrication and software development enable insight into the details of design and implementation. These insights often translate to increased assurance.

  Acquired hardware and software elements may not provide the opportunity to achieve the same insight into design and implementation and may offer more functionality and capability than required. The limits of what can be known about the internals of the elements translate to a level of uncertainty about vulnerability and the maximum assurance that can be achieved.
- *Firmware:* Firmware exhibits properties of hardware and software. Firmware elements may be acquired or developed to realize the software aspects and then fabricated to realize the physical form of the hardware aspects. Firmware elements, therefore, adhere to the security implementation considerations of both hardware and software elements.
- *Services:* System elements implemented by obtaining or leasing services are subject to the same criteria used to acquire hardware, firmware, and software but must also address the security considerations associated with utilization and support resources.
- *Utilization and Support Resources:* The security considerations of acquired or leased services account for the specific roles and responsibilities of individuals of the service/lease provider

and their ability to account for all of the security requirements and constraints associated with the delivery, utilization, and sustainment of the service or capability being leased.

IP-2.2 Place the system element in a secure state for future use, as needed.

Note: This includes protection of the element while stored and in transit, as well as the packaging and labeling of the element.

IP-2.3 Record objective evidence that system elements meet the system security requirements.

References: [4] [30] [61] [86] [98] [99] [109] [111] [112] [113]

IP-3 MANAGE RESULTS OF IMPLEMENTATION

IP-3.1 Record the security aspects of implementation results and any anomalies encountered.

IP-3.2 Maintain traceability of the security aspects of implemented system elements.

Note: Bidirectional traceability of the security aspects of the implemented system elements to the system security requirements, the security views of the architecture, the security design, and the security interface requirements is maintained.

IP-3.3 Provide the security-relevant artifacts that have been selected for baselines.

References: [4] [30] [61] [98] [99]

H.8. Integration

The purpose of the *Integration* process is to synthesize a set of system elements into a realized system that satisfies the system requirements.

H.8.1. Security Purpose

- Synthesize a set of system elements into a realized trustworthy secure system that satisfies the system requirements.

H.8.2. Security Outcomes

- Security-relevant integration constraints that influence requirements, architecture, design, or interfaces and interactions are identified.
- Approaches and checkpoints for the correct secure activation of the identified interfaces and system functions to an initial or established secure state are developed.
- Enabling systems or services for the security aspects of integration are available.
- A trustworthy secure system composed of implemented system elements is integrated.
- Security aspects of system external interfaces (system to external environment) and system internal interfaces (between implemented system elements) are checked.
- Security aspects of integration results and anomalies are identified.

- Traceability of the security aspects of the integrated system elements is established.

H.8.3. Security Activities and Tasks

IN-1 PREPARE FOR INTEGRATION

IN-1.1 Identify and define checkpoints for the correct secure activation and integrity of the interfaces and the selected system functions as the system elements are synthesized.

IN-1.2 Define the security aspects of the integration strategy.

Note: Integration is performed to achieve trustworthy secure results using aspects such as secure assembly sequences and checkpoints for the system elements based on established priorities while minimizing integration time and cost and providing appropriate risk treatments.

IN-1.3 Identify the security-relevant constraints and objectives from integration to be incorporated into the system requirements, architecture, or design.

IN-1.4 Identify the security aspects for enabling systems, services, and materials needed to support integration.

IN-1.5 Identify and plan for enabling systems, services, and materials needed to support the security aspects of integration.

IN-1.6 Obtain or acquire access to the security aspects of enabling systems, services, and materials to be used in integration.

References: [4] [30] [61] [81] [98] [99] [100] [111] [112] [113]

IN-2 PERFORM INTEGRATION

IN-2.1 Check interface availability and conformance of the interfaces in accordance with the security aspects of interface definitions and integration schedules.

IN-2.2 Perform actions to address any security-relevant conformance or availability issues.

IN-2.3 Securely combine the implemented system elements in accordance with planned sequences.

IN-2.4 Securely integrate system element configurations until the complete system is securely synthesized.

IN-2.5 Check for the expected results of interfaces, interconnections, selected functions, and security characteristics.

References: [4] [86] [100] [109] [111] [112] [113]

IN-3 MANAGE RESULTS OF INTEGRATION

IN-3.1 Record the security aspects of integration results and any anomalies encountered.

Note: Anomaly analyses determine corrective actions that may affect the protection capability of the system and the level of assurance that can be obtained.

IN-3.2 Maintain traceability of the security aspects of integrated system elements.

Note: Bidirectional traceability of the security aspects of the integrated system elements to the system security requirements, security views of the architecture, security design, and security

interface requirements is maintained. Traceability provides evidence that supports assurance and trustworthiness claims.

IN-3.3 Provide the security-relevant artifacts that have been selected for baselines.

References: [4] [30] [61] [98] [99] [100]

H.9. Verification

The purpose of the *Verification* process is to provide objective evidence that a system, system element, or artifact fulfills its specified requirements and characteristics.

H.9.1. Security Purpose

- Provide objective evidence that a system, system element, or artifact (e.g., system requirements, architecture description, or design description) fulfills its specified security requirements and characteristics.
- Identify security-relevant anomalies[98] in any artifact, implemented system elements, or life cycle processes, and provide the necessary information to determine the resolution of such anomalies.

H.9.2. Security Outcomes

- Security-relevant verification constraints that influence requirements, architecture, or design are identified.
- Enabling systems or services for the security aspects of verification are available.
- Security aspects of the system, system element, or artifact are verified.
- Security-relevant data that provides information for corrective actions is reported.
- Objective evidence that the realized system fulfills the security requirements and security aspects of the architecture and design is provided.
- Security aspects of verification results and anomalies are identified.
- Traceability of the security aspects of the verified system elements is established.

H.9.3. Security Activities and Tasks

VE-1 PREPARE FOR VERIFICATION

VE-1.1 Identify the security aspects within the verification scope and corresponding security verification actions.

Note: Scope includes the system, system elements, information items, or artifacts that will be verified against applicable requirements, security characteristics, or other security properties. Each verification action description includes what will be verified (e.g., actual system, model,

[98] Anomalies include behaviors and outcomes that are observed but not specified.

mock-up, prototype, procedure, plan, or other information item), the verification method (including any adversity emulation), and the expected result as defined by the success criteria. The security criteria may reflect considerations of strength of function/mechanism, resistance to tamper, misuse or abuse, penetration resistance, level of assurance, absence of flaws, weaknesses, and the absence of unspecified behavior and outcomes.

VE-1.2 Identify the constraints that can potentially limit the feasibility of the security-focused verification actions.

Note: Constraints include technical feasibility; the availability of qualified personnel and verification enablers; the availability of sufficient, relevant, and credible threat data; technology employed (including adversity emulation); the size and complexity of the system element or artifact; and the cost and time allotted for the verification.

VE-1.3 Select appropriate security verification methods and the associated success criteria for each security verification action.

Note: The methods and techniques are selected to provide the evidence required to achieve the expected results with the desired level of assurance.

VE-1.4 Define the security aspects of the verification strategy.

Note: This includes the approach used to incorporate security considerations into all verification actions while considering the trade-offs between scope, depth, and rigor needed for the desired level of assurance and the given constraints.

VE-1.5 Identify the security-relevant constraints and objectives that result from the security aspects of the verification strategy to be incorporated into the system requirements, architecture, and design.

VE-1.6 Identify the security aspects for enabling systems or services needed to support verification.

VE-1.7 Identify and plan for enabling systems or services needed to support the security aspects of verification.

VE-1.8 Obtain or acquire access to the security aspects of enabling systems or services to be used in verification.

References: [4] [30] [31] [61] [86] [[97] [98] [99] [100] [119] [120] [121] [122]

VE-2 PERFORM VERIFICATION

VE-2.1 Define the security aspects of the verification procedures, each supporting one or a set of verification actions.

Note: The procedures identify the security purpose of verification, the success criteria (expected results), the verification method to be applied, the necessary enabling systems (e.g., facilities, equipment), and the environmental conditions to perform each verification procedure (e.g., resources, qualified personnel, adversity emulations).

VE-2.2 Perform security verification procedures.

References: [4] [86] [97] [100] [109]

VE-3 MANAGE RESULTS OF VERIFICATION

VE-3.1 Record the security aspects of verification results and any anomalies encountered.

VE-3.2 Obtain agreement from the approval authority that the system, system element, or artifact meets the specified system security requirements.

Note: There may be multiple approval authorities with security-relevant responsibilities.

VE-3.3 Maintain traceability of the security aspects of verification.

Note: Bidirectional traceability is maintained between the verified security aspects of system elements and the system security requirements, architecture, design, and interface requirements. This traceability includes verification results or evidence, such as security-relevant anomalies, deviations, or requirement satisfaction.

VE-3.4 Provide the security-relevant artifacts that have been selected for baselines.

References: [4] [30] [61] [98] [99] [100] [109]

H.10. Transition

The purpose of the *Transition* process is to establish a capability for a system to provide services specified by stakeholder requirements in the operational environment.

H.10.1. Security Purpose

- Preserve the system's verified security characteristics during the orderly and planned transition of the system to be operable in the intended environment, which may be a new or changed environment.

H.10.2. Security Outcomes

- Security-relevant transition constraints that influence system requirements, architecture, or design are identified.
- Enabling systems or services for the security aspects of transition are available.
- The prepared site satisfies security criteria.
- The system is installed in its operational environment and can deliver its specified functions in a trustworthy secure manner.
- Operators, users, and other stakeholders necessary to the system utilization and support are trained in the system's security capabilities, mechanisms, and features.
- Security-relevant transition results and anomalies are identified.
- The installed system is activated and ready for trustworthy secure operation.
- Traceability of the security aspects of the transitioned elements is established.

H.10.3. Security Activities and Tasks

TR-1 PREPARE FOR TRANSITION

TR-1.1 Define the security aspects of the transition strategy.

Note: The transition strategy includes all security-relevant activities, from site delivery and installation through the deployment and commissioning of the system, as well as all security-relevant stakeholders, including human operators. The strategy also includes security roles and responsibilities, facility security considerations, secure shipping and receiving, contingency back out plans, security training, security aspects of installation acceptance demonstration tasks, secure operational readiness reviews, secure operations commencement, transition security success criteria, rights of secure access, data rights, and integration with other plans. System commissioning is considered along with the secure decommissioning of the old system when one exists. In this case, the Transition and Disposal processes are used concurrently.

TR-1.2 Identify and define any security-relevant facility or site changes needed.

TR-1.3 Identify the security-relevant constraints and objectives from the security aspects of transition to be incorporated into the system requirements, architecture, and design.

TR-1.4 Identify and arrange the security training of operators, users, and other stakeholders necessary to the system utilization and support.

TR-1.5 Identify the security aspects for enabling systems or services needed to support transition.

TR-1.6 Identify and plan for enabling systems or services needed to support the security aspects of transition.

TR-1.7 Obtain or acquire access to the security aspects of enabling systems or services to be used in transition.

TR-1.8 Identify security aspects, and arrange for the secure shipping and receiving of system elements and enabling systems.

References: [4] [30] [61] [98] [99]

TR-2 PERFORM TRANSITION

TR-2.1 Prepare the site of operation in accordance with secure installation requirements.

TR-2.2 Securely deliver the system for installation at the correct location and time.

Note: Secure delivery considers the various forms, means, and methods that accomplish the end-to-end transport of system elements to ensure that they are not tampered with during transport. Items and packages are delivered only to the intended recipient, which may mean shipping with more lead time to account for additional security.

TR-2.3 Install the system in its operational environment in accordance with the secure installation strategy, and establish secure interconnections to its environment.

TR-2.4 Demonstrate trustworthy secure system installation.

Note: The installation and connection procedures are to be properly verified to provide confidence that the intended system configuration across all system modes and states is

achieved. This includes completing acceptance tests defined in agreements. These tests include security aspects associated with physical connections between the system and the environment.

TR-2.5 Provide security training for the operators, users, and other stakeholders necessary for system utilization and support.

TR-2.6 Perform security activation and checkout of the system.

Note: Security activation and checkout shows that the system can initialize to its initial secure operational state for all defined modes of operation and accounts for all interconnections to other systems across physical, virtual, and wireless interfaces.

TR-2.7 Demonstrate that the installed system can deliver its required functions in a trustworthy secure manner.

TR-2.8 Demonstrate that the security functions provided by the system and the effects of the security functions are sustainable by enabling systems.

TR-2.9 Review the security trustworthiness of the system for operational readiness.

Note: The results of installation, operational, and enabling system checkouts are reviewed to determine whether the security performance and effectiveness are sufficient to justify operational use.

TR-2.10 Commission the system for secure operation.

Note: This includes providing security support to users and operators at the time of the system commissioning.

References: [4] [86]

TR-3 MANAGE RESULTS OF TRANSITION

TR-3.1 Record the security aspects of transition results and any anomalies encountered.

TR-3.2 Record the security aspects of operational incidents and problems, and track their resolution.

TR-3.3 Maintain traceability of the security aspects of transitioned system elements.

Note: Bidirectional traceability is maintained between all identified security aspects and the supporting data associated with the transition strategy and the system requirements, system architecture, and system design.

TR-3.4 Provide the security-relevant artifacts that have been selected for baselines.

References: [4]

H.11. Validation

The purpose of the *Validation* process is to provide objective evidence that the system, when in use, fulfills its business or mission objectives and stakeholder requirements, achieving its intended use in its intended operational environment.

H.11.1. Security Purpose

Provide objective evidence that the system, when in use, fulfills the protection needs associated with its business or mission objectives and the stakeholder requirements, thereby achieving its intended use in its intended operational environment in a trustworthy secure manner.

H.11.2. Security Outcomes

- Security validation criteria are defined.
- The availability of security services required by stakeholders is confirmed.
- Security-relevant validation constraints that influence system requirements, architecture, or design are identified.
- Security aspects of the system, system element, or artifact are validated.
- Enabling systems or services for the security aspects of validation are available.
- Security-focused validation results and anomalies are identified.
- Objective evidence of the successful validation of security aspects is provided.
- Traceability of the validated security aspects of the system, system elements, and artifacts is established.

H.11.3. Security Activities and Tasks

VA-1 PREPARE FOR VALIDATION

VA-1.1 Identify the security aspects within the validation scope and corresponding security validation actions.

Note: The security aspects of validation focus on the stakeholders' protection needs, concerns, and associated security requirements. The scope includes system elements, the entire system, or any artifact that impacts the stakeholder's confidence in the system and the decision to accept the system as being trustworthy for its intended use.

VA-1.2 Identify the constraints that can potentially limit the feasibility of the security validation actions.

Note: Constraints may include the level of assurance and the availability of business or mission stakeholders to support validation activities; the availability of sufficient, relevant, and credible threat data; the limits on conducting validation activities in actual operational conditions across all business and mission modes and associated system states and modes; the technology employed; the size and complexity of the system element or artifact; and the cost and time allotted for validation activities.

VA-1.3 Select appropriate security validation methods and the associated success criteria for each security validation action.

Note: Adversity emulation, including penetration testing and emulating abuse and misuse, is included.

VA-1.4 Develop the security aspects of the validation strategy.

Note: The security aspects of the validation strategy address the approach to incorporate security considerations into all validation actions, while considering the trade-offs between scope, depth, and rigor needed for the desired level of assurance and the given constraints.

VA-1.5 Identify the security-relevant system constraints that result from the security aspects of the validation strategy to be incorporated in the stakeholder protection needs and the requirements transformed from those needs.

Note: These constraints are associated with the clarity and accuracy of the expression of needs and requirements to achieve the desired level of assurance with certainty and repeatability.

VA-1.6 Identify the security aspects for enabling systems or services needed to support validation.

VA-1.7 Identify and plan for enabling systems or services to support the security aspects of validation.

VA-1.8 Obtain or acquire access to the security aspects of enabling systems or services to be used to support validation.

References: [4] [30] [61] [86] [97] [98] [99] [100] [118]

VA-2 PERFORM VALIDATION

VA-2.1 Define the security aspects of the validation procedures, each supporting one or a set of validation actions.

Note: This includes identification of the validation methods or techniques to be employed, the qualifications of the individuals conducting the validation, and any specialized equipment that may be needed, such as what may be required to emulate environmental adversities.

VA-2.2 Perform security validation procedures.

Note 1: Security-focused validation actions from the execution of validation procedures contribute to demonstrating that the system is sufficiently trustworthy.

Note 2: The performance of a security-focused validation action consists of capturing a result from the execution of the procedure, comparing the obtained result with the expected result, deducing the degree of compliance of the element, and deciding about the acceptability of compliance if uncertainty remains.

References: [4] [86] [97] [100] [118]

VA-3 MANAGE RESULTS OF VALIDATION

VA-3.1 Record the security aspects of validation results and any anomalies encountered.

Note: The recorded validation results include nonconformance issues, anomalies, or problems that are potentially security related. These results inform the analyses to determine causes and enable corrective or improvement actions. Corrective actions may affect the security aspects of the system architecture definition, design definition, system security requirements and associated constraints, the level of assurance that can be obtained, and/or the implementation strategy, including its security aspects.

VA-3.2 Record the security characteristics of operational incidents and problems, and track their resolution.

Note: Incidents that occur in the operational environment of the system are recorded and subsequently correlated to validation activities and results. This is an important feedback loop for continuous improvement in the engineering of trustworthy secure systems.

VA-3.3 Obtain agreement that the security validation criteria have been met.

VA-3.4 Maintain traceability of the security aspects of validation.

Note: Bidirectional traceability of the security aspects of validated system elements to stakeholder protection needs, security concerns, and security requirements is maintained. Traceability demonstrates completeness of the validation process and provides evidence that supports assurance and trustworthiness claims.

VA-3.5 Provide the security-relevant artifacts that have been selected for baselines.

References: [4] [30] [61] [98] [99] [100]

H.12. Operation

The purpose of the *Operation* process is to use the system to deliver its services.

H.12.1. Security Purpose

- Inform the security aspects of the requirements and constraints to securely operate the system and monitor the security aspects of products, services, and operator-system performance.
- Identify and analyze security-relevant operational anomalies.

H.12.2. Security Outcomes

- Security aspects of operation constraints that influence system requirements, architecture, or design are identified.
- Enabling systems, services, and material for the security aspects of operation are available.
- Trained and qualified personnel who can securely operate the system are available.
- System products or services that meet stakeholder security requirements are delivered.
- Security aspects of system performance during operation are monitored.
- Security support is provided to stakeholders.

H.12.3. Security Activities and Tasks

OP-1 PREPARE FOR OPERATION

OP-1.1 Define the security aspects of the operation strategy.

Note 1: This includes the approach to enable the continuous secure operation and use of the system and its security services, as well as the provision of support to operations elements to address anomalies identified during the operation and use of the system. It also includes:

- The capacity, availability, schedule considerations, and security of products or services as they are introduced, routinely operated, and disposed (including contingency operations)
- The human resources strategy and security qualification requirements for personnel, including all associated security-relevant training and personnel compliance requirements
- The security aspects of release and re-acceptance criteria and schedules of the system to permit modifications that sustain the security aspects of existing or enhanced products or services
- The approach to implement operational modes in the System Operational Concept, including normal and contingency operations
- The secure approaches for contingency, degraded, alternative, training, and other modes of operation, as well as the transition within and between modes while considering resilience in the face of adversity
- Measures for operation that will provide security insights into performance levels
- The approach to achieving situational awareness to determine security-relevant consequences

Note 2: This includes planning for securely starting the system, halting the system, shutting the system down, operating the system in a training mode, the secure implementation of work-around procedures to restore operations, performing back-out and restore operations, operating in any degraded mode, or alternative modes for special conditions. If needed, the operator performs the necessary steps to enter into contingency operations and possibly power down the system. Contingency operations are performed in accordance with the pre-established procedures for such an event.

Note 3: There may be a need to plan for certain modes of operation for which security functions and services are reduced or eliminated to achieve more critical system functions and services or to carry out certain maintenance or periodic testing. Predetermined procedures for entering and exiting such modes would be followed.

OP-1.2 Identify the constraints and objectives that result from the security aspects of operation to be incorporated into the system requirements, architecture, and design.

OP-1.3 Identify the security aspects for enabling systems and services needed to support operation.

OP-1.4 Identify and plan for enabling systems or services needed to support the security aspects of operation.

OP-1.5 Obtain or acquire access to the security aspects of enabling systems or services to be used in operation.

OP-1.6 Identify or define security training and qualification requirements to sustain the workforce needed for secure system operation.

Note: Security qualification and training includes role and function-oriented competency, proficiency, certification, and other criteria to securely operate and use the system in all of its defined modes or states.

OP-1.7 Assign the trained and qualified personnel needed for secure system operation.

References: [4] [86] [100] [111] [112] [113] [114]

OP-2 PERFORM OPERATION

OP-2.1 Securely use the system in its intended operational environment.

OP-2.2 Apply materials and other resources as required to securely operate the system and sustain its product and service capabilities.

Note 1: Materials and resources are provided by logistical actions. Logistics is discussed as part of the maintenance process.

Note 2: Operational personnel may perform system modification and support activities, such as software updates.

OP-2.3 Monitor system operations for deviations from intended behavior and outcomes.

Note: This includes (1) managing adherence to the operation strategy and operational procedures (the operations conducted by personnel), (2) monitoring that the system is operated in a secure manner and compliant with regulations, procedures, and directives, and (3) monitoring for anomalies that may not be directly observable as system behavior and may or may not be obviously security relevant.

OP-2.4 Use the measures defined in the strategy and analyze them to confirm that system security performance is within acceptable parameters.

Note: System monitoring includes reviewing whether the performance is within established security-relevant thresholds (loss margins), periodic instrument readings are acceptable, and service and response times are acceptable. Operator feedback and suggestions are useful input for improving the security aspects of system operational performance.

OP-2.5 Identify and record when system security or service performance is not within acceptable parameters.

References: [4] [30] [61] [86] [98] [99] [100]

OP-3 MANAGE RESULTS OF OPERATION

OP-3.1 Record the results of secure operations and any anomalies encountered.

Note: Anomalies include those associated with the operation strategy, the operation of enabling systems, the execution of the operation, and incorrect system definition, all of which may be due to security issues or may result in security issues.

OP-3.2 Record the security aspects of operational incidents and problems, and track their resolution.

OP-3.3 Maintain traceability of the security aspects of the operation elements.

OP-3.4 Provide the security-relevant artifacts that have been selected for baselines.

References: [4] [30] [61] [98] [99] [100]

OP-4 SUPPORT STAKEHOLDERS

OP-4.1 Provide security assistance and consultation to stakeholders as requested.

Note: Assistance and consultation includes the provision or recommendation of sources for security-relevant training, security aspects of documentation, vulnerability resolution, security reporting, and other security-relevant support services that enable the effective and secure use of the product or service.

OP-4.2 Record and monitor requests and subsequent actions for security support.

OP-4.3 Determine the degree to which the security aspects of delivered products and services satisfy the needs of stakeholders.

References: [4] [86] [100]

H.13. Maintenance

The purpose of the *Maintenance* process is to sustain the capability of the system to provide a product or service.

H.13.1. Security Purpose

- Establish the security aspects of requirements and constraints to securely sustain the capability of the system to provide a product or service.

  Note: Secure sustainment includes all maintenance and logistics activities for the packaging, handling, storage, and transportation of replacement system elements.

H.13.2. Security Outcomes

- Security aspects of maintenance and logistics constraints that influence system requirements, architecture, or design are identified.
- Enabling systems or services needed for the security aspects of system maintenance and logistics are available.
- Replacement, repaired, or modified system elements are securely made available.
- The need for required security-relevant maintenance and logistics actions is reported.
- Security-relevant failures and life cycle data, including associated costs, are determined.

H.13.3. Security Activities and Tasks

MA-1 PREPARE FOR MAINTENANCE AND LOGISTICS

MA-1.1 Define the security aspects of the maintenance strategy.

Note: The maintenance strategy seeks to preserve the secure capability and performance of the delivered system. The security aspects of the maintenance strategy generally include:

- The secure transition of the system and system elements into a secure maintenance mode or state, as well as the secure transition back to operation

- An approach to ensure that sourced materials and system elements that do not meet specified quality, origin, and functionality (e.g., counterfeit) are not introduced into the system
- The skill and personnel levels required to effect repairs, replacements, and restoration while accounting for maintenance staff requirements and any relevant legislation regarding health, safety, security, and the environment
- Maintenance measures that provide insight into the security aspects of performance levels, effectiveness, and efficiency

MA-1.2 Define the security aspects of the logistics strategy.

Note: The logistics strategy defines the specific security considerations required to perform logistics throughout the life cycle. This generally includes:

- Acquisition logistics to help ensure that security implications are considered early during the development stage
- Operations logistics to help ensure that the necessary material and resources are securely made available in the right quantity and quality and at the right place and time; considerations for securely making material and resources available include identification and marking, packaging, distribution, handling, and provisioning
- The security criteria for storage locations and conditions, as well as the number and type of replacement system security-specific elements, their anticipated replacement rate, and their storage life and renewal frequency

MA-1.3 Identify the security-relevant constraints and objectives that result from the security aspects of maintenance and logistics to be incorporated into the system requirements, architecture, and design.

MA-1.4 Identify trade-offs such that the security aspects of the system and associated maintenance and logistics actions result in a solution that is trustworthy, secure, affordable, operable, supportable, and sustainable.

Note: The cost of secure maintenance and logistics should be considered within the lifetime cost of the system.

MA-1.5 Identify the security aspects for enabling systems, products, and services needed to support maintenance and logistics.

MA-1.6 Identify and plan for enabling systems, products, and services needed to support the security aspects of maintenance and logistics.

MA-1.7 Obtain or acquire access to the security aspects of enabling systems, products, and services to be used in maintenance and logistics.

References: [4] [30] [61] [86] [98] [99] [100] [111] [112] [113] [114] [115]

MA-2 PERFORM MAINTENANCE

Note: The need to perform maintenance may be driven by the need to address explicit security issues, incidents, or failures. All maintenance actions must be accomplished in a secure manner with the understanding that some actions may have a direct effect on the security posture of the system.

MA-2.1 Monitor and review stakeholder requirements and incident and problem reports to identify security-relevant corrective, preventive, adaptive, additive, or perfective maintenance needs.

Note: Security-relevant maintenance needs include those needs that are direct (e.g., an identified security incident) or indirect (e.g., considerations to securely address a maintenance need).

MA-2.2 Record the security aspects of maintenance incidents and problems, and track their secure resolution.

MA-2.3 Analyze the impact of the changes introduced by maintenance actions on the security aspects of the system and system elements.

MA-2.4 Upon encountering faults that cause a system failure, securely restore the system to secure operational status.

Note: Secure restoration means that the maintenance action itself does not worsen the secure state or condition of the system.

MA-2.5 Securely correct anomalies (e.g., defects, errors, and faults), and replace or upgrade system elements.

MA-2.6 Perform preventive maintenance by securely replacing or servicing system elements prior to failure.

MA-2.7 Securely perform adaptive, additive, or perfective maintenance as required.

References: [4] [30] [61] [86] [98] [99] [100] [114] [115]

MA-3 PERFORM LOGISTICS SUPPORT

MA-3.1 Perform the security aspects of acquisition logistics.

MA-3.2 Perform the security aspects of operational logistics.

MA-3.3 Implement mechanisms for the secure logistics needed during the life cycle.

Note 1: These mechanisms enable secure packaging, handling, storage, and transportation.

Note 2: These mechanisms aid in the prevention and detection of counterfeits, tampering, substitution, and redirection.

MA-3.4 Confirm that the security aspects of logistics actions are implemented.

Note: The security aspects of logistics actions satisfy both logistics protection concerns and the need to meet repair rates, replenishment levels, and planned schedules.

References: [4] [30] [61] [98] [99] [100] [111] [112] [113] [114] [115]

MA-4 MANAGE RESULTS OF MAINTENANCE AND LOGISTICS

MA-4.1 Record the security aspects of maintenance and logistics results and any anomalies encountered.

MA-4.2 Record maintenance and logistics security incidents and problems, and track their secure resolution.

MA-4.3 Identify and record the security-relevant trends of incidents, problems, and maintenance and logistics actions.

MA-4.4 Maintain traceability of the security aspects of maintenance and logistics.

MA-4.5 Provide security-relevant artifacts that have been selected for baselines.

MA-4.6 Monitor customer satisfaction with the security aspects of the system, maintenance, and logistics.

References: [4] [30] [61] [98] [99] [100] [114] [115] [116]

H.14. Disposal

The purpose of the *Disposal* process is to end the existence of a system element or system for a specified intended use, appropriately handle replaced or retired elements and any waste products, and properly attend to identified critical disposal needs (e.g., per an agreement, per organizational policy, or for environmental, legal, safety, or security aspects).

H.14.1. Security Purpose

- Provide the aspects needed to securely end the existence of a system element or system for a specified use, and securely preserve or destroy the associated data and information.

H.14.2. Security Outcomes

- Secure disposal constraints that influence system requirements, architecture, design, and implementation are identified.
- Enabling systems or services for the security aspects of disposal are available.
- System elements are destroyed, stored, reclaimed, or recycled in accordance with safety and security requirements.
- The environment is securely returned to its original secure or an agreed-upon secure state.
- Records of the security aspects of disposal actions and analysis are available.

H.14.3. Security Activities and Tasks

DS-1 PREPARE FOR DISPOSAL

DS-1.1 Define the security aspects of the disposal strategy.

Note: The security aspects address securely terminating system functions and services, transforming the system and environment into an acceptable secure state, addressing security concerns, and transitioning the system and system elements for future use. The disposal strategy determines approaches, schedules, resources, specific considerations of secure disposal, and the effectiveness and completeness of secure disposal and disposition actions.

- *Permanent termination of system functions and delivery of services:* The security aspects address the removal, decommissioning, or destruction of the associated system elements while preserving the security posture of any remaining functions and services.
- *Transform the system and environment into an acceptable state:* The security aspects address any alterations made to the system, its operation, and the environment to ensure that stakeholder protection needs and concerns are addressed by the remaining portions of the system and the functions and services it provides. When the entire system is removed, the security aspects address alterations to the environment to return it to its original or agreed-upon secure state.
- *Address security concerns for material, data, and information:* The security aspects address protections for sensitive components, technology, data, and information removed from service, dismantled, stored, prepared for reuse, or destroyed. The aspects may include the duration of protection level/state, downgrades, releasability, and criteria that define authorized access and use during the storage period. The protection needs for disposal are defined by stakeholders and agreements and may be subject to regulatory requirements, expectations, and constraints.
- *Transition the system and system elements for future use:* The security aspects address the transition of the system or system elements for future use in a modified or adapted form, including legacy migration and return to service. The security aspects may include constraints, limitations, or other criteria to enable recovery of the systems' functions and services within a specified time or to ensure security-oriented interoperability with future enabling systems and other systems. These aspects may also include periodic inspections to account for the security posture and return-to-service readiness of stored system elements, associated data and information, and all supporting operations and sustainment support materials. The security aspects apply to all system functions and services and are not limited to only security protection-oriented functions and services of the system.

DS-1.2 Identify the security-relevant constraints and objectives of disposal on the system requirements, architecture and design characteristics, and implementation techniques.

DS-1.3 Identify the security aspects for enabling systems or services needed to support disposal.

DS-1.4 Identify and plan for enabling systems or services needed to support the security aspects of disposal.

DS-1.5 Obtain or acquire access to the security aspects of enabling systems or services to be used in disposal.

DS-1.6 Specify security criteria for containment facilities, storage locations, inspection, and storage periods (if the system is to be stored).

DS-1.7 Define the security aspects of preventive methods to preclude disposed elements and materials that should not be repurposed, reclaimed, or reused from re-entering the supply chain.

References: [4] [86] [100]

DS-2 PERFORM DISPOSAL

DS-2.1 Securely deactivate the system or system element to prepare it for secure removal from operation.

Note: Deactivation is accomplished to preserve the security posture of the system.

DS-2.2 Securely remove the system, system element, or waste material from use or production for appropriate secure disposition and action.

DS-2.3 Securely withdraw impacted operating staff from the system or system element, and record relevant secure operation knowledge.

DS-2.4 Securely disassemble the system or system element into manageable elements to facilitate its secure removal for reuse, recycling, reconditioning, overhaul, archiving, or destruction.

Note: Secure disassembly preserves the security characteristics of the system elements that are not removed.

DS-2.5 Securely handle system elements and their parts that are not intended for reuse in a manner that will help ensure that they do not get back into the supply chain.

DS-2.6 Conduct secure sanitization and destruction of the system elements and life cycle artifacts.

Note 1: Governing agreements, laws, and regulations determine the appropriate means to sanitize and destroy data, information, and system elements that contain data and information, as well as retention periods before sanitization and destruction can occur.

Note 2: Sanitization and destruction techniques include clearing, purging, cryptographic erase, physical modification, and physical destruction.

Note 3: Sanitization and destruction techniques and methods may be specific to data, information, and system element type.

References: [4] [86] [100]

DS-3 FINALIZE THE DISPOSAL

DS-3.1 Confirm that no detrimental security factors exist following disposal.

DS-3.2 Return the environment to its original secure state or to a secure state specified by agreement.

DS-3.3 Securely archive data and information gathered through the lifetime of the system to permit audits and reviews in the event of long-term hazards to health, safety, security, and the environment and to permit future system creators and users to securely build a knowledge base from past experiences.

DS-3.4 Provide security-relevant artifacts that have been selected for baselines.

References: [4] [100]

Appendix I. Technical Management Processes

This appendix contains the *Technical Management Processes* from [4] with security-relevant considerations and contributions for the purpose, outcomes, activities, and tasks.

I.1. Project Planning

The purpose of the *Project Planning* process is to produce and coordinate effective and workable plans.

I.1.1. Security Purpose

- Determine and coordinate the security aspects of effective and workable plans.

I.1.2. Security Outcomes

- Security objectives, security-specific plans, and security aspects of other plans are defined.
- Security-relevant roles, responsibilities, accountabilities, and authorities within the project are defined.
- Security aspects of performance and achievement criteria are defined.
- The resources and services necessary to achieve the security objectives are committed.
- Plans for the execution of the security aspects of the project are activated.

I.1.3. Security Activities and Tasks

PL-1 DEFINE THE PROJECT

PL-1.1 Identify the security aspects of project objectives and constraints.

Note: Objectives and constraints include strategic security, assurance, and trustworthiness goals, as well as loss thresholds and regulatory concerns. Each security-relevant objective is identified with a level of detail that permits selecting, tailoring, and implementing the appropriate processes and activities.

PL-1.2 Define the security aspects of the project scope as established in agreements.

Note: This includes the relevant activities required to satisfy the security aspects of decision criteria and complete the project successfully.

PL-1.3 Define and maintain the security views of the project life cycle model that are comprised of stages using the defined life cycle models of the organization.

PL-1.4 Establish appropriate security aspects of the work breakdown structure.

Note: Each security-relevant element of the work breakdown structure is described with a level of detail that is consistent with identified security risks and required visibility.

PL-1.5 Define and maintain the security aspects of processes that will be applied to the project.

Note: Entry criteria, inputs, process sequence constraints, and Measures of Effectiveness and/or Measures of Performance attributes may all have security aspects.

References: [4][30] [61] [86] [96] [98] [99] [100] [111] [112] [113] [123]

PL-2 PLAN PROJECT AND TECHNICAL MANAGEMENT

PL-2.1 Define and maintain the security aspects of a project schedule based on management and technical objectives, and work estimates.

Note: This includes security aspects that impact the definition of the duration, relationship, dependencies, and sequence of activities; achievement milestones; resources employed; reviews (including security subject matter expertise employed); and schedule reserves for security risk management necessary to achieve timely completion of the project.

PL-2.2 Define the security aspects of achievement criteria for the life cycle decision gates, delivery dates, and major dependencies on external inputs and outputs.

Note: This includes the criteria defined by regulatory, certification, evaluation, and other approval authorities.

PL-2.3 Define the security aspects of project performance criteria.

PL-2.4 Define the security-relevant project costs, and plan the budget.

PL-2.5 Define the security-relevant roles, responsibilities, accountabilities, and authorities.

Note: This includes defining the project organization, staff acquisitions, and development of staff security-relevant skills. Authorities include legally responsible roles and individuals, as appropriate. These security-relevant authorities include security design authorization, security test and operation authorization, and the award of certification, accreditation, or authorization.

PL-2.6 Define the security aspects of the infrastructure and services required.

Note: This includes defining the capacity needed for security infrastructure and services, its availability, and its allocation to project tasks. Security infrastructure includes facilities (e.g., Sensitive Compartmented Information Facilities [SCIFs] and isolated networks), specific strength of mechanism mediated access, cross-domain solutions, tools, communication, and information technology assets.

PL-2.7 Plan the security aspects of acquiring materials and enabling system services supplied from outside of the project.

PL-2.8 Generate and communicate a plan for the security aspects of project and technical management and execution, including security reviews that address security considerations.

Note: Security considerations and the planning to address those considerations are captured in a Systems Engineering Management Plan, Software Engineering Management Plans, and similar plans.

References: [4] [30] [61] [86] [98] [99] [100] [111] [112] [113]

PL-3 ACTIVATE THE PROJECT

PL-3.1 Obtain authorization for the security aspects of the project.

PL-3.2 Submit requests and obtain commitments for the necessary resources to perform the security aspects of the project.

PL-3.3 Implement the security aspects of project plans.

References: [4] [86] [100]

I.2. Project Assessment and Control

The purpose of the *Project Assessment and Control* process is to assess if the plans are aligned and feasible; determine the status of the project, technical, and process performance; and direct execution to help ensure that the performance is within projected budgets according to plans and schedules to satisfy technical objectives.

I.2.1. Security Purpose

- Assess whether the security aspects of plans and security plans are aligned and feasible.
- Determine the state of the project, technical, and process security performance.
- Direct execution to help ensure that the security performance is within projected budgets according to plans and schedules to satisfy security and other technical objectives.

I.2.2. Security Outcomes

- Security aspects of performance measures or assessment results are available.
- Security-relevant roles, responsibilities, accountabilities, authorities, and resources are assessed for adequacy.
- Security aspects of technical progress reviews are performed.
- Deviations in the security aspects of project performance from plans are analyzed.
- Affected stakeholders are informed of the security aspects of the project's status.
- Corrective action is directed when project performance or achievement does not meet security-relevant targets.
- Security aspects of project replanning are initiated, as necessary.
- Security aspects of project action to progress (or not) from one scheduled milestone or event to the next are authorized.

I.2.3. Security Activities and Tasks

PA-1 PLAN FOR PROJECT ASSESSMENT AND CONTROL

PA-1.1 Define the security aspects of the project assessment and control strategy.

Note 1: This includes the planned security assessment methods and time frames, as well as necessary security management and technical reviews.

Note 2: Expectations of regulatory, certification, and authorization entities inform the security aspects of the project assessment and control strategy.

References: [4] [30] [61] [98] [99] [100]

PA-2 ASSESS THE PROJECT

PA-2.1 Assess the alignment of the security aspects of project objectives and plans with the project context.

PA-2.2 Assess the security aspects of the management and technical plans against objectives to determine adequacy and feasibility.

PA-2.3 Assess the security aspects of the project and technical status against appropriate plans to determine actual and projected cost, schedule, and performance variances.

PA-2.4 Assess the adequacy of the security-relevant roles, responsibilities, accountabilities, and authorities.

Note: This includes assessment of the adequacy of personnel competencies to perform project roles and accomplish project tasks.

PA-2.5 Assess the security aspects of resource adequacy and availability.

PA-2.6 Assess progress using measured security achievement and the security aspects of milestone completion.

Note: This includes collecting and evaluating security-relevant data for labor, materials, service costs, and technical performance, as well as other technical data about security objectives. These are compared against security-relevant measures of achievement, including conducting effectiveness assessments to determine the adequacy of the evolving system to fulfill security requirements.

PA-2.7 Conduct required management and technical reviews, audits, and inspections relevant to the security aspects of the project.

Note: The reviews, audits, and inspections are formal or informal and are conducted to determine the security-relevant readiness to proceed to the next stage or milestone, to help ensure that project and technical security objectives are being meet, or to solicit feedback from stakeholders with security concerns.

PA-2.8 Monitor the security aspects of critical processes and new technologies.

Note: This includes identifying and evaluating technology maturity from a security perspective, as well as the feasibility of technology insertion for satisfying security objectives.

PA-2.9 Make recommendations based on security measurement results and other security-relevant project information.

Note: Measurement results are analyzed to identify security-relevant deviations, variations, or undesirable trends from planned values and to make security-relevant recommendations for corrective, preventive, adaptive, additive, or perfective actions.

PA-2.10 Record and provide security status and security findings from the assessment tasks.

PA-2.11 Monitor the security aspects of process execution within the project.

Note: This includes an analysis of process security measures and a review of security-relevant trends with respect to project objectives.

References: [4] [30] [61] [86] [98] [99] [100]

PA-3 CONTROL THE PROJECT

PA-3.1 Initiate the actions needed to address identified security issues.

PA-3.2 Initiate the necessary security aspects of project replanning.

Note: Replanning is initiated when the security aspects of project objectives or constraints have changed or when security-relevant planning assumptions are shown to be invalid.

PA-3.3 Initiate necessary change actions when there is a contractual change to cost, time, or quality due to the security impact of an acquirer or supplier request.

Note: The security impact is not necessarily obvious when the request is not security-driven or security-oriented.

PA-3.4 Recommend that the project proceed toward the next milestone or event, if justified, based on the achievement of security-relevant milestones or event criteria.

References: [4] [86] [100] [111] [112] [113]

I.3. Decision Management

The purpose of the *Decision Management* process is to provide a structured, analytical framework for objectively identifying, characterizing, and evaluating a set of alternatives for a decision at any point in the life cycle and select the most beneficial course of action.

I.3.1. Security Purpose

- Identify, analyze, characterize, and evaluate the security aspects of alternatives for a decision.
- Recommend the most beneficial course of security-informed action.

I.3.2. Security Outcomes

- Security aspects of decisions requiring alternative analysis are identified.
- Security aspects of alternative courses of action are identified and evaluated.
- A preferred security-informed course of action is selected.
- Security aspects of a resolution, the decision rationale, and the assumptions are identified.

I.3.3. Security Activities and Tasks

DM-1 PREPARE FOR DECISIONS

DM-1.1 Define the security aspects of the decision management strategy.

Note: A decision management strategy includes the identification of security-relevant roles, responsibilities, accountabilities, and authorities. It also includes the identification of security-specific decision categories and a prioritization scheme. Security-relevant decisions often arise because of a security effectiveness assessment, a technical trade-off, a security-relevant problem that needs to be solved, a response to a security risk that exceeds the acceptable threshold, or a new opportunity.

DM-1.2 Identify the security aspects of the circumstances and the need for a decision.

DM-1.3 Identify stakeholders with relevant security expertise to support decision-making efforts.

References: [4] [86] [100]

DM-2 ANALYZE THE DECISION INFORMATION

DM-2.1 Select and declare the security aspects of the decision management strategy for each decision.

Note: This includes the security-relevant level of rigor and the data and system analysis needed.

DM-2.2 Determine the desired security outcomes and the measurable security attributes of selection criteria.

Note: The desired value for all quantifiable security criteria and the threshold value(s) beyond which the attribute will be unsatisfactory are determined.

DM-2.3 Identify the security aspects of the trade space and alternatives.

Note: If many alternatives exist, security aspects are to qualitatively screen to reduce alternatives to a manageable number for further detailed system analysis.

DM-2.4 Evaluate each alternative against the security criteria.

References: [4] [86] [100]

DM-3 MAKE AND MANAGE DECISIONS

DM-3.1 Determine the preferred alternative for each security-informed and security-based decision.

DM-3.2 Record the security-informed or security-based resolution, decision rationale, and assumptions.

DM-3.3 Record, track, evaluate, and report the security aspects of security-informed and security-based decisions.

Note: Security aspects of problems or opportunities and the alternative courses of action that will resolve their outcomes – including those with security impacts – are recorded, categorized, and reported.

References: [4] [86] [100]

I.4. Risk Management

The purpose of the *Risk Management* process is to identify, analyze, treat, and monitor the risks continually.

I.4.1. Security Purpose

- Continually identify, analyze, treat, and monitor the risks associated with the uncertainty of achieving security objectives and the effects of security protection efforts on achieving system objectives.

I.4.2. Security Outcomes

- Security-relevant risks are identified.
- Security-relevant risks are analyzed.
- Security-relevant risk treatments are selected.
- Security-relevant risk treatments are implemented.
- Security-relevant risks are evaluated on an ongoing basis to assess changes in status and progress in treatment.
- Security-relevant risks are recorded and maintained in the risk profile.

I.4.3. Security Activities and Tasks

RM-1 PLAN RISK MANAGEMENT

RM-1.1 Define the security aspects of the risk management strategy.

Note 1: The nature of security risk includes intentional and unintentional casual events, considerations of the intended behaviors and outcomes, functions (security and other functions), and the potential effects of security risk realization. Casual events may be combinations of events in the operational environment and events in the system environment.

Note 2: The security aspects scope of the risk management process, risk management approach, risk criteria, measures, parameters, rating scale, and treatment alternatives are defined. This includes security aspects of the risk management process at all levels of the supply chain (e.g., suppliers, subcontractors) and how they are incorporated into the project risk management process.

Note 3: The strategy can include those security-relevant issues (i.e., risks with a likelihood of occurrence of 1) and opportunities (i.e., risks with positive outcomes) within scope and approach. Opportunity aspects include opportunity criteria, measures, parameters, rating scale, and treatment alternatives.

RM-1.2 Define and record the security context of the risk management process.

Note 1: This includes the identification of security-relevant stakeholders and descriptions of their perspectives, risk categories, and technical and managerial objectives, assumptions, and constraints.

Note 2: Security opportunities provide potential benefits for the system or project. Security contexts consider the security impact of not pursuing an opportunity and the security risk of not achieving the effects provided by the opportunity.

References: [4] [30] [61] [86] [98] [99] [100] [124] [125]

RM-2 MANAGE THE RISK PROFILE

RM-2.1 Define and record the security risk thresholds and conditions.

Note: The security risk thresholds define the levels at which the appropriate treatment strategies are considered.

RM-2.2 Establish and maintain the security aspects of the risk profile.

Note: The risk profile records each security risk and opportunity, including a description of the security risk or opportunity, a record of the risk or opportunity parameters, the priority based on the risk or opportunity criteria, and the risk or opportunity's current state, treatment, and contingency strategy. The risk profile is updated when an individual security risk or opportunity state changes.

RM-2.3 Provide the security aspects of the relevant risk profile to stakeholders.

Note: Project planning determines the frequency of communicating the risk profile and its security aspects.

References: [4] [86] [100] [124] [125]

RM-3 ANALYZE RISK

RM-3.1 Identify security risks in the categories described in the risk management context.

Note: Security risks are commonly identified through various security and other analyses (e.g., safety, assurance, producibility, and performance analyses); technology, architecture, integration, and readiness assessments; measurement reports; and trade-off studies. Additionally, security risks are often identified through the analysis of measures associated with system security goals (e.g., security-relevant Measures of Effectiveness or Measures of Performance).

RM-3.2 Measure each identified security risk.

RM-3.3 Evaluate each security risk against its risk thresholds.

RM-3.4 Define and record recommended treatment strategies and measures for each security-relevant risk that exceeds its risk threshold.

References: [4] [30] [61] [86] [98] [99] [100] [124] [125]

RM-4 TREAT RISKS THAT EXCEED THEIR RISK THRESHOLD

RM-4.1 Identify recommended alternatives for security risk treatment.

RM-4.2 Define measures for determining the effectiveness of security risk treatments.

RM-4.3 Implement selected security risk treatments.

Note: The implemented alternative should be the one for which the security-relevant stakeholders determine that the actions taken will make a security-relevant risk acceptable.

RM-4.4 Coordinate management action for selected security risk treatments.

References: [4] [86] [100] [124] [125]

RM-5 MONITOR RISK

RM-5.1 Continually monitor all security-relevant risks and the security risk management context.

Note: Changes with security-relevant risks and their treatments may prompt reevaluation. The initial treatment plans for a security-relevant risk may include preplanned additional actions when risk increases or insufficiently decreases despite treatment.

RM-5.2 Implement and monitor measures to evaluate the effectiveness of security-relevant risk treatments.

RM-5.3 Continually monitor for the emergence of new security-relevant risks and sources of risk throughout the life cycle.

Note: This includes monitoring known changes in adversities.

References: [4] [30] [61] [86] [98] [99] [100] [124] [125]

I.5. Configuration Management

The purpose of the *Configuration Management* process is to manage system and system elements and configurations over the life cycle.

I.5.1. Security Purpose

- Incorporate security considerations to securely manage system and system elements and configurations over the life cycle.

I.5.2. Security Outcomes

- System element configurations are securely managed.
- Security aspects of configuration baselines are established.
- Changes to items under configuration management are securely controlled.
- Security aspects of configuration status information are available.
- Security aspects of required configuration audits are completed.
- Security aspects of system releases are approved.

I.5.3. Security Activities and Tasks

CM-1 PREPARE FOR CONFIGURATION MANAGEMENT

CM-1.1 Define a secure configuration management strategy.

Note: These include:

- Security-relevant roles, responsibilities, accountabilities, and authorities

- Criteria for the secure management of changes to items under configuration management, including dispositions, access, release, and control
- Security considerations, criteria, and constraints for the locations, conditions, and environment of storage
- Criteria or events for commencing secure configuration control and securely maintaining the baselines of evolving configurations
- Security aspects of the audit strategy and the responsibilities for assessing the continual integrity and security of the configuration definition information
- Criteria and constraints for secure change management, planned configuration control boards and security configuration control boards, regulatory and emergency change requests, and procedures for secure change management
- Secure coordination among stakeholders, acquirers, suppliers, supply chain, and other interacting organizations

CM-1.2 Define the secure archive and retrieval approach for configuration items, configuration management artifacts, and data.

Note: This includes rules that govern secure retention, access, and use.

References: [4] [74] [86] [100] [126] [127]

CM-2 PERFORM CONFIGURATION IDENTIFICATION

CM-2.1 Identify the security aspects of system elements and artifacts that need to be under configuration management.

CM-2.2 Identify the security aspects of the configuration data to be managed.

CM-2.3 Establish the security aspects of identifiers for items under configuration management.

CM-2.4 Define the security aspects of baselines throughout the life cycle.

CM-2.5 Obtain applicable stakeholder agreement on the security aspects to establish a baseline.

CM-2.6 Approve and track the security aspects of system or system element releases.

Note 1: The security aspects of a release are security-relevant considerations of authorization of the use of a system or system element for a specific purpose with or without security-relevant restrictions. Examples are releases for tests or operational use.

Note 2: Releases generally include a set of changes made through the Technical Processes. Release approval generally includes acceptance of the verified and validated changes and any impacts to security of the changes.

References: [4] [86] [100] [111] [112] [113]

CM-3 PERFORM CONFIGURATION CHANGE MANAGEMENT

CM-3.1 Identify and record the security aspects of requests for change and requests for variance.

Note 1: This includes requests for deviation, waiver, or concession.

Note 2: Change or variance can be based on reasons other than security or without an obvious relevance to security.

CM-3.2 Determine the security aspects of action to coordinate, evaluate, and disposition requests for change or requests for variance.

Note: The security aspects identified are coordinated and evaluated across all impacted performance and effectiveness evaluation criteria, as well as the criteria of project plans, cost, benefits, risks, quality, and schedule.

CM-3.3 Submit requests for security review and approval.

Note: Control boards may or may not be security focused. For a non-security control board activity, security should be reviewed to verify that a request has no security aspects.

CM-3.4 Track and manage the security aspects of approved changes to the baseline, requests for change, and requests for variance.

References: [4] [86] [100]

CM-4 PERFORM CONFIGURATION STATUS ACCOUNTING

CM-4.1 Develop and maintain security-relevant configuration management status information for system elements, baselines, approved changes, and releases.

Note: The information includes security certification, accreditation, authorization, or approval decisions for a system, system element, baseline, or release.

CM-4.2 Capture, store, and report security-relevant configuration management data.

References: [4] [86] [100]

CM-5 PERFORM CONFIGURATION EVALUATION

CM-5.1 Identify the need for secure configuration and configuration management verification activities and audits.

CM-5.2 Verify that the product or service configuration meets the security-relevant configuration requirements.

Note: This is performed by comparing security requirements, constraints, and waivers (variances) with the results of formal verification activities.

CM-5.3 Monitor the secure incorporation of approved configuration changes.

CM-5.4 Perform configuration and configuration management security verification activities and audits to establish the security aspects of product baselines.

Note: This includes the security aspects of the functional configuration audit (FCA) that are focused on functional and performance capabilities and of the physical configuration audit (PCA) that are focused on system conformance to operational and configuration information items.

CM-5.5 Record the security aspects of the configuration management audit and other configuration evaluation results and disposition action items.

References: [4] [86] [100]

I.6. Information Management

The purpose of the *Information Management* process is to generate, obtain, confirm, transform, retain, retrieve, disseminate, and dispose of information to designated stakeholders.

I.6.1. Security Purpose

- Address the security aspects of information management.

I.6.2. Security Outcomes

- Security-relevant information to be managed is identified.
- Security protections for information are identified.
- Security aspects of information representations are defined.
- Information is securely managed.
- Security aspects of information status are identified.
- Information is available to designated stakeholders in a secure manner.

I.6.3. Security Activities and Tasks

IM-1 PREPARE FOR INFORMATION MANAGEMENT

IM-1.1 Define the security aspects of the strategy for information management.

Note: The security aspects include stakeholder, technical, and other information. These aspects address security, privacy, and intellectual property concerns.

IM-1.2 Define the security aspects of the information items that will be managed.

IM-1.3 Designate authorities and responsibilities for the security aspects of information management.

Note: Due regard is paid to legislation, security, and privacy (e.g., ownership, agreement restrictions, rights of access, data rights, and intellectual property). Where restrictions or constraints apply, information is identified accordingly. Staff with knowledge of such items of information are informed of their security-relevant obligations and responsibilities.

IM-1.4 Define the security aspects of the content, formats, structure, and strengths of protection for information items.

Note 1: The security aspects apply to information while at rest (i.e., persistent or non-persistent storage), while in transit between a source/point of origin and destination, and while in transformation.

Note 2: The security aspects are informed by the criteria in applicable laws, policies, directives, regulations, and patents.

IM-1.5 Define the security aspects of information maintenance actions.

References: [4] [86] [100] [128]

IM-2 PERFORM INFORMATION MANAGEMENT

IM-2.1 Securely obtain, develop, or transform the identified information items.

Note: Obtaining, developing, and transforming information items includes labeling the items by their protection needs (e.g., classifying).

IM-2.2 Securely maintain information items and their storage records, and record the security status of information.

IM-2.3 Securely publish, distribute, or provide access to information and information items to designated stakeholders.

IM-2.4 Securely archive designated information.

Note: The media, location, and protection of the information are selected in accordance with the specified storage and retrieval periods, agreements, legislation, and organizational security policy.

IM-2.5 Securely dispose of unwanted, invalid, or unvalidated information.

References: [4] [86] [100] [128] [129]

I.7. Measurement

The purpose of the *Measurement* process is to collect, analyze, and report objective data and information to support effective management and demonstrate the quality of the products, services, and processes.

I.7.1. Security Purpose

- Collect, analyze, and report security-relevant data and information to support effective management and demonstrate the quality of the products, services, and processes.

I.7.2. Security Outcomes

- Security-relevant information needs are identified.
- An appropriate set of security measures are identified or developed based on security-relevant information needs and information security protection needs.
- Required data is securely managed.
- Security-relevant data is analyzed, and the results are interpreted.
- Measurement results provide objective information that supports security-relevant decisions.

I.7.3. Security Activities and Tasks

MS-1 PREPARE FOR MEASUREMENT

MS-1.1 Define the security aspects of the measurement strategy.

MS-1.2 Describe the characteristics of the organization that are relevant to security measurement.

MS-1.3 Identify and prioritize security-relevant information needs.

Note: The needs are based on protection objectives, risks, and other security-relevant items related to project decisions.

MS-1.4 Select and specify measures that satisfy security-relevant information needs.

MS-1.5 Define procedures for the collection, analysis, access, and reporting of security-relevant data.

MS-1.6 Define security-relevant criteria for evaluating the information items and the measurement process.

Note: All criteria for a security-relevant information item are security-relevant.

MS-1.7 Identify the security aspects for enabling the systems or services needed to support measurement.

MS-1.8 Identify and plan for enabling the systems or services needed to support the security aspects of measurement.

MS-1.9 Obtain or acquire access to the security aspects of enabling systems or services to be used in measurement.

References: [4] [86] [130] [79] [97]

MS-2 PERFORM MEASUREMENT

MS-2.1 Integrate procedures for the generation, collection, analysis, and reporting of security-relevant data into the relevant processes.

MS-2.2 Integrate procedures for the secure generation, collection, analysis, and reporting of data into the relevant processes.

MS-2.3 Collect, store, and verify security-relevant data.

MS-2.4 Securely collect, store, and verify data.

MS-2.5 Analyze security-relevant data, and develop security-relevant information items.

MS-2.6 Record security measurement results, and inform the measurement users.

Note: Security measurement results are provided to stakeholders and project personnel to support decision-making and risk management and to initiate corrective actions and improvements.

References: [4] [79] [86] [97] [130]

I.8. Quality Assurance

The purpose of the *Quality Assurance* process is to help ensure the effective application of the organization's Quality Management process to the project.

I.8.1. Security Purpose

- Ensure the effective application of the organization's Quality Management process to the security aspects of the project.

  Note: The security aspects for Quality Assurance should account for the assurance tenets detailed in Appendix F.2.

I.8.2. Security Outcomes

- Security aspects of quality assurance procedures, including security criteria and methods for quality assurance evaluations, are implemented.
- Evaluations of the products, services, and processes of the project are performed in a manner consistent with security quality management policies, procedures, and requirements.
- Security results of evaluations are provided to relevant stakeholders.
- Security-relevant incidents are resolved.
- Prioritized security-relevant problems are treated.

I.8.3. Security Activities and Tasks

QA-1 PREPARE FOR QUALITY ASSURANCE

QA-1.1 Define the security aspects of the quality assurance strategy.

Note: The security aspects are informed by and consistent with the quality management policies, objectives, and procedures and include:

- Project security quality assurance procedures
- Security roles, responsibilities, accountabilities, and authorities
- Security activities appropriate to each life cycle process
- Security activities appropriate to each supplier (including subcontractors)
- Required security-oriented verification, validation, monitoring, measurement, inspection, and test activities specific to the product or service
- Security criteria for product or service acceptance

QA-1.2 Establish the independence of security quality assurance from other life cycle processes.

References: [4] [30] [61] [86] [98] [99] [106] [107] [108] [131]

QA-2 PERFORM PRODUCT OR SERVICE EVALUATIONS

QA-2.1 Evaluate products and services for conformance to established security criteria, contracts, standards, and regulations.

QA-2.2 Perform the security aspects of verification and validation on the outputs of the life cycle processes to determine conformance to specified requirements.

References: [4] [30] [61] [86] [98] [99] [131]

QA-3 PERFORM PROCESS EVALUATIONS

QA-3.1 Evaluate project life cycle processes for conformance to established security quality criteria.

QA-3.2 Evaluate tools and environments that support or automate the process for conformance to established security quality criteria.

QA-3.3 Evaluate supplier processes for conformance to process security requirements.

Note: Consider items such as the security aspects of development environments, process measures required of suppliers, or risk processes that suppliers are required to use.

References: [4] [30] [61] [86] [98] [99] [111] [112] [113] [131]

QA-4 MANAGE QUALITY ASSURANCE RECORDS AND REPORTS

QA-4.1 Create records and reports related to the security aspects of quality assurance activities.

QA-4.2 Securely maintain, store, and distribute records and reports.

QA-4.3 Identify the security aspects of incidents and problems associated with product, service, and process evaluations.

References: [4] [30] [61] [86] [98] [99] [131]

QA-5 TREAT INCIDENTS AND PROBLEMS

QA-5.1 Record, analyze, and classify the security aspects of incidents.

Note: Incidents are grouped (classified) by criteria such as type, scope, and effect.

QA-5.2 Resolve the security aspects of incidents or elevate the security aspects of incidents to problems.

QA-5.3 Record, analyze, and classify the security aspects of problems.

QA-5.4 Track the security aspects of the prioritization and implementation of problem treatments.

Note: This includes both security-driven problem treatment and the security aspects of general problem treatments.

QA-5.5 Note and analyze the security aspects of incidents and problems.

QA-5.6 Inform stakeholders of the status of the security aspects of incidents and problems.

QA-5.7 Track the security aspects of incidents and problems to closure.

References: [4] [30] [61] [98] [99] [131]

Appendix J. Organizational Project-Enabling Processes

This appendix contains the *Organizational Project-Enabling Processes* from [4] with security-relevant considerations and contributions for the purpose, outcomes, activities, and tasks.

J.1. Life Cycle Model Management

The purpose of the *Life Cycle Model Management* process is to define, maintain, and help ensure the availability of policies, life cycle processes, life cycle models, and procedures for use by the organization with respect to the scope of this International Standard.

J.1.1. Security Purpose

- Ensure that security needs and considerations are incorporated into the policies, life cycle processes, life cycle models, and procedures used by the organization.

J.1.2. Security Outcomes

- Security considerations are captured in organizational policies and procedures for the management and deployment of life cycle models and processes.
- Security roles, responsibilities, accountabilities, and authorities within life cycle policies, processes, models, and procedures are defined.
- The selection of policies, life cycle processes, life cycle models, and procedures for use by the organization is informed by security needs and considerations.
- Security needs and considerations for policies, life cycle processes, life cycle models, and procedures for use by the organization are assessed.
- Prioritized security-relevant process, model, and procedure improvements are implemented.

J.1.3. Security Activities and Tasks

LM-1 ESTABLISH THE LIFE CYCLE PROCESSES

LM-1.1 Establish policies and procedures for process management and deployment that are consistent with the security aspects of organizational strategies.

Note: The policies and procedures may be security-focused, security-based, or may have security-informing aspects.

LM-1.2 Establish the security aspects of the life cycle processes that implement the requirements of [4] and are consistent with organizational strategies.

LM-1.3 Define the security roles, responsibilities, accountabilities, and authorities to facilitate implementation of the security aspects of life cycle processes and the strategic management of life cycles.

LM-1.4 Define the security aspects of the criteria that control progression through the life cycle.

Note: This includes security criteria for gates, checkpoints, and entry/exit criteria for milestones and decision points.

LM-1.5 Establish security criteria for the standard life cycle models for the organization, including criteria for outcomes for each stage.

Note: The life cycle model comprises one or more stages, as needed, with each stage having security aspects to its purpose and outcomes. The model is assembled as a sequence of stages that overlap or iterate as appropriate for the scope of the system of interest, magnitude, complexity, changing needs, and opportunities (including protection needs and opportunities).

References: [4] [30] [61] [86] [98] [99] [100] [132]

LM-2 ASSESS THE LIFE CYCLE PROCESS

LM-2.1 Monitor the security aspects of process execution across the organization.

Note: This includes the analysis of process measures and the review of security-relevant trends with respect to strategic security criteria, feedback from projects regarding the effectiveness and efficiency of the processes, and monitoring execution according to regulations and organizational policies.

LM-2.2 Conduct reviews of the security aspects of the life cycle models used by the projects.

Note: This includes confirming the suitability, adequacy, and effectiveness of the life cycle models used by the project. The reviews should be conducted periodically and be event-driven (e.g., at completions of large project milestones).

LM-2.3 Identify security-relevant improvement opportunities from assessment results.

References: [4] [30] [61] [86] [98] [99] [100] [134]

LM-3 IMPROVE THE PROCESS

LM-3.1 Prioritize and plan for security-relevant improvement opportunities.

LM-3.2 Implement security improvement opportunities, and inform relevant stakeholders.

Note: This includes regulatory, certification, accreditation, acceptance, and similar stakeholders.

References: [4] [30] [61] [86] [98] [99] [100]

J.2. Infrastructure Management

The purpose of the *Infrastructure Management* process is to provide infrastructure and services to projects to support organization and project objectives throughout the life cycle.

J.2.1. Security Purpose

- Define the protection needs for the aspects of infrastructure and services that support organization and project objectives.

J.2.2. Security Outcomes

- Protection needs for the infrastructure are defined.
- Security capabilities and constraints of infrastructure elements are specified.
- Infrastructure elements that satisfy infrastructure security specifications are obtained.
- Secure infrastructure is available.
- Prioritized infrastructure security-relevant improvements are implemented.

J.2.3. Security Activities and Tasks

IF-1 ESTABLISH THE INFRASTRUCTURE

IF-1.1 Define the infrastructure security protection needs.

Note: The security aspects of infrastructure resource needs are considered in context with other projects and resources within the organization. Security constraints that influence and control the provision of infrastructure resources and services for the project are also defined.

IF-1.2 Identify, obtain, and provide the infrastructure resources and services that satisfy the security protection needs to securely implement and support projects.

References: [4] [30] [61] [86] [98] [99] [100] [111] [112] [113]

IF-2 MAINTAIN THE INFRASTRUCTURE

IF-2.1 Evaluate the degree to which delivered infrastructure resources satisfy project protection needs.

IF-2.2 Identify and provide security improvements or changes to infrastructure resources as project requirements change.

Note: Any mismatch between project security needs and the security provided by infrastructure resources may result in gaps in assurance.

References: [4] [30] [61] [86] [98] [99] [100] [111] [112] [113]

J.3. Portfolio Management

The purpose of the *Portfolio Management* process is to initiate and sustain necessary, sufficient, and suitable projects in order to meet the strategic objectives of the organization.

J.3.1. Security Purpose

- Identify security considerations for projects to meet the strategic objectives of the organization.

J.3.2. Security Outcomes

- Security aspects of strategic venture opportunities, investments, or necessities are prioritized.

- Security aspects of projects are identified.
- Resources and budgets for the security aspects of each project are allocated.
- Project management responsibilities, accountabilities, and authorities for security are defined.
- Projects that meet the security criteria in agreements and stakeholder security requirements are sustained.
- Projects that do not meet the security criteria in agreements or do not satisfy stakeholder security requirements are redirected or terminated.
- Projects that have completed the security aspects of agreements and that satisfy stakeholder security requirements are closed.

J.3.3. Security Activities and Tasks

PM-1 DEFINE AND AUTHORIZE PROJECTS

PM-1.1 Identify potential new or modified security capabilities or missions.

Note: The organization strategy, concept of operations, or gap or opportunity analysis is reviewed to identify security-driven gaps, problems, or opportunities.

PM-1.2 Identify the security aspects of potential new or modified capabilities or missions.

Note: The organization strategy, concept of operations, or gap or opportunity analysis is reviewed to identify security-relevant gaps, problems, or opportunities.

PM-1.3 Prioritize, select, and establish new business opportunities, ventures, or undertakings with consideration for security objectives and concerns.

PM-1.4 Define the security aspects of projects, accountabilities, and authorities.

Note: This includes project proprietary, sensitivity, and privacy criteria.

PM-1.5 Identify the security aspects of the expected goals, objectives, and outcomes of each project.

Note: This includes project proprietary, sensitivity, and privacy criteria.

PM-1.6 Identify and allocate resources for achieving the security aspects of project goals and objectives.

PM-1.7 Identify the security aspects of any multi-project interfaces and dependencies to be managed or supported by each project.

Note: This includes interfaces and dependencies with enabling systems and services, as well as all associated data and information.

PM-1.8 Specify the security aspects of project reporting requirements, and review milestones that govern the execution of each project.

PM-1.9 Authorize each project to commence the execution of project plans, including its security aspects.

References: [4] [30] [61] [86] [98] [99] [100]

PM-2 EVALUATE THE PORTFOLIO OF PROJECTS

PM-2.1 Evaluate the security aspects of projects to confirm ongoing viability.

Note: This includes the following:

- The project is progressing toward achieving established security goals and objectives.
- The project is complying with project security directives.
- The project is being conducted according to the security aspects of project life cycle policies, processes, and procedures.
- The project remains viable, as indicated by the continuing need for security services, practical secure product implementation, and acceptable security-driven investment benefits.

PM-2.2 Act to continue projects that are satisfactorily progressing in consideration of project security aspects.

PM-2.3 Act to redirect projects that can be expected to progress satisfactorily with appropriate security-informed redirection.

References: [4] [86] [100]

PM-3 TERMINATE PROJECTS

PM-3.1 Where agreements permit, act to cancel or suspend projects whose security-driven disadvantages or security-driven risks to the organization outweigh the benefits of continued investments.

PM-3.2 After completion of the agreement for the security aspects of products or services, act to close the projects.

Note: Closure is accomplished in accordance with organizational security policies, procedures, and the agreement.

References: [4] [86] [100]

J.4. Human Resource Management

The purpose of the *Human Resource Management* process is to provide the organization with necessary human resources and to maintain their competencies in a manner consistent with strategic needs.

J.4.1. Security Purpose

- Define the security criteria for necessary human resources, and maintain their competencies in a manner consistent with strategic needs.

J.4.2. Security Outcomes

- Security-relevant skills required by projects are identified.

  Personnel with necessary security skills are provided to projects.

- Security-relevant skills of personnel are developed, maintained, or enhanced.
- Security-relevant personnel conflicts are resolved.

J.4.3. Security Activities and Tasks

HR-1 IDENTIFY SKILLS

HR-1.1 Identify the security-relevant skills needed based on current and expected projects.

HR-1.2 Identify and record the security-relevant skills of personnel.

References: [4] [86] [100] [132] [109] [133]

HR-2 DEVELOP SKILLS

HR-2.1 Establish a plan for developing security-relevant skills.

Note: Security-relevant skills include core and specialty competencies.

HR-2.2 Obtain security-relevant training, education, or mentoring resources.

HR-2.3 Provide planned security-relevant skills development.

HR-2.4 Maintain records of security-relevant skills development.

References: [4] [86] [100] [109] [132]

HR-3 ACQUIRE AND PROVIDE SKILLS

HR-3.1 Obtain qualified personnel when security-relevant skill deficits are identified.

HR-3.2 Maintain and manage the pool of security-skilled personnel necessary to staff ongoing projects.

HR-3.3 Make personnel assignments based on security-relevant project and staff development needs.

HR-3.4 Motivate personnel with security-relevant skills (e.g., through career development and reward mechanisms).

HR-3.5 Resolve the security aspects of personnel conflicts across or within projects.

Note: Conflicts across or within projects may include personnel capacity, availability, qualification conflicts, and personality conflicts.

References: [4] [86] [133]

J.5. Quality Management

The purpose of the *Quality Management* process is to assure that products, services, and implementations of the quality management process meet organizational and project quality objectives and achieve customer satisfaction.

J.5.1. Security Purpose

- Define organizational and project security quality objectives and the criteria used to determine that products, services, and implementations of the Quality Management process meet those security objectives.

  Note: The security aspects for Quality Management should account for the assurance tenets detailed in Appendix F.2.

J.5.2. Security Outcomes

- Organizational security quality management policies, standards, and procedures are defined and implemented.
- Security quality evaluation criteria and methods are established.
- Resources and information are provided to projects to support the operation and monitoring of project security quality assurance activities.
- Security aspects of quality evaluation results are analyzed.
- Security quality management policies and procedures are improved based on project and organization results.

J.5.3. Security Activities and Tasks

QM-1 PLAN QUALITY MANAGEMENT

QM-1.1 Establish the security aspects of quality management policies, standards, and procedures.

QM-1.2 Define responsibilities and authority for the implementation of security quality management.

QM-1.3 Define security quality evaluation criteria and methods.

QM-1.4 Provide resources, data, and information for security quality management.

References: [4] [30] [61] [86] [98] [99] [130]

QM-2 ASSESS QUALITY MANAGEMENT

QM-2.1 Gather and analyze quality assurance evaluation results in accordance with the defined security quality evaluation criteria.

QM-2.2 Assess customer satisfaction.

QM-2.3 Conduct periodic reviews of project quality assurance activities for compliance with the security quality management policies, standards, and procedures.

QM-2.4 Monitor the status of security quality improvements on processes, products, and services.

References: [4] [30] [61] [86] [98] [99] [130]

QM-3 PERFORM QUALITY MANAGEMENT CORRECTIVE AND PREVENTIVE ACTIONS

QM-3.1 Plan corrective actions when security quality management objectives are not achieved.

QM-3.2 Plan preventive actions when there is a sufficient risk that security quality management objectives will not be achieved.

QM-3.3 Monitor the security aspects of corrective and preventive actions to completion, and inform stakeholders.

References: [4] [30] [61] [86] [98] [99] [130]

J.6. Knowledge Management

The purpose of the Knowledge Management process is to create the capability and assets that enable the organization to exploit opportunities to reapply existing knowledge.

J.6.1. Security Purpose

- Enable the organization to exploit opportunities to reapply existing security knowledge.

J.6.2. Security Outcomes

- A taxonomy for the application of security-relevant knowledge assets is identified.
- Organizational security knowledge, skills, and knowledge assets are organized.
- Organizational security knowledge, skills, and knowledge assets are available.
- Organizational security knowledge, skills, and knowledge assets are communicated across the organization.
- Security knowledge management usage data is analyzed.

J.6.3. Security Activities and Tasks

KM-1 PLAN KNOWLEDGE MANAGEMENT

KM-1.1 Define the security aspects of the knowledge management strategy.

Note: The security aspects of the knowledge management strategy include:

- The security knowledge domains and technologies and their potential for the reapplication of knowledge
- The plans for obtaining and maintaining security knowledge, skills, and security knowledge assets for their useful life
- The types of security knowledge, security skills, and security knowledge assets to be collected and maintained
- The criteria for accepting, qualifying, and retiring security knowledge, security skills, and security knowledge assets

- The procedures for controlling changes to the security knowledge, security skills, and security knowledge assets
- The plans, mechanisms, and procedures for protection, control, and access to classified or sensitive data and information
- The mechanisms for secure storage and secure retrieval

KM-1.2 Identify the security knowledge, skills, and knowledge assets to be managed.

KM-1.3 Identify projects that can benefit from the application of the security knowledge, skills, and knowledge assets.

References: [4] [86] [100] [132] [133]

KM-2 SHARE KNOWLEDGE AND SKILLS THROUGHOUT THE ORGANIZATION

KM-2.1 Establish and maintain a classification for capturing and sharing security knowledge and skills.

Note: This classification includes security expert, common security, and security domains knowledge and skills, as well as lessons learned.

KM-2.2 Capture or acquire security knowledge and skills.

KM-2.3 Make security knowledge and skills accessible across the organization.

References: [4] [86] [100]

KM-3 SHARE KNOWLEDGE ASSETS THROUGHOUT THE ORGANIZATION

KM-3.1 Establish a taxonomy to organize security knowledge assets.

Note: The taxonomy includes the following:

- Definition of the boundaries of security domains and their relationships to one another
- Definition of the boundaries of security-relevant domains (e.g., safety) and their relationships to one another
- Domain models that capture essential common and different security-relevant features, capabilities, concepts, and functions

KM-3.2 Develop or acquire security knowledge assets.

Note: Security knowledge assets include system elements or their representations (e.g., reusable code libraries, security reference architectures), architecture or design elements (e.g., security architecture or security design patterns), processes, security criteria, or other technical information (e.g., training materials) related to security domain knowledge and lessons learned.

KM-3.3 Make all knowledge assets securely accessible to the organization.

References: [4] [71] [86] [100]

KM-4 MANAGE KNOWLEDGE, SKILLS, AND KNOWLEDGE ASSETS

KM-4.1 Maintain security knowledge, skills, and knowledge assets.

KM-4.2 Monitor and record the use of security knowledge, skills, and knowledge assets.

KM-4.3 Periodically reassess the currency of the security aspects of technology and market needs of the security knowledge assets.

References: [4] [86] [100]

Appendix K. Agreement Processes

This appendix contains the *Agreement Processes* from [4] with security-relevant considerations and contributions for the purpose, outcomes, activities, and tasks.

K.1. Acquisition

The purpose of the *Acquisition* process is to obtain a product or service in accordance with the acquirer's requirements.

K.1.1. Security Purpose

- Obtain a product or service in accordance with the acquirer's security requirements.

K.1.2. Security Outcomes

- A request for supply includes security criteria.
- One or more suppliers are selected that satisfy the security criteria.
- An agreement containing security criteria is established between the acquirer and supplier.
- A product or service that complies with the security criteria in the agreement is accepted.
- The security aspects of the acquirer obligations defined in the agreement are satisfied.

K.1.3. Security Activities and Tasks

AQ-1 PREPARE FOR THE ACQUISITION

AQ-1.1 Define the security aspects of the strategy for how the acquisition will be conducted.

Note: This strategy describes or references the life cycle model, security risks and issues mitigation, a schedule of security-relevant milestones, the protection of acquirer and supplier assets, and security-relevant selection criteria if the supplier is external to the acquiring organization. It also includes key security drivers and security-relevant characteristics of the acquisition, such as responsibilities and liabilities; specific models, methods, or processes; formality; level of criticality; and the priority of security within relevant trade-off factors.

AQ-1.2 Prepare a request for a product or service that includes the security requirements.

Note: The request includes security criteria for the business practices with which the supplier is to comply, a list of bidders with adequate security qualifications, and the security criteria that will be used to select the supplier.

References: [4] [30] [61] [86] [98] [99] [100] [111] [112] [113]

AQ-2 ADVERTISE THE ACQUISITION AND SELECT THE SUPPLIER

AQ-2.1 Securely communicate the request for the supply of a product or service to potential suppliers.

AQ-2.2 Select one or more suppliers that meet the security criteria.

References: [4] [30] [61] [86] [98] [99] [100] [111] [112] [113]

AQ-3 ESTABLISH AND MAINTAIN AN AGREEMENT

AQ-3.1 Develop and approve an agreement with the supplier that includes security acceptance criteria.

Note: This agreement can range in formality. Appropriate to the level of formality, the agreement establishes security requirements, secure development and delivery milestones, security verification, security validation, and the security aspects of acceptance conditions, process requirements (e.g., configuration management, risk management, and measurement), and the handling of data rights and intellectual property. The security aspects of the agreement also include the application of all of the above to subcontractors and other supporting organizations to the supplier.

AQ-3.2 Identify necessary security-relevant changes to the agreement.

AQ-3.3 Evaluate the security impact of changes to the agreement.

Note: The basis for the agreement change may or may not be security-related. However, there may be a security-relevant impact regardless of the basis for the change.

AQ-3.4 Update the security criteria in the agreement with the supplier, as necessary.

References: [4] [30] [61] [86] [98] [99] [100] [111] [112] [113]

AQ-4 MONITOR THE AGREEMENTS

AQ-4.1 Assess the execution of the security aspects of the agreement.

Note: This includes confirmation that all parties are meeting their security-relevant responsibilities according to the agreement.

AQ-4.2 Securely provide data needed by the supplier, and resolve issues in a timely manner.

References: [4] [86] [100] [111] [112] [113]

AQ-5 ACCEPT THE PRODUCT OR SERVICE

AQ-5.1 Confirm that the delivered product or service complies with the security aspects of the agreement.

AQ-5.2 Securely provide payment or other agreed consideration.

AQ-5.3 Accept the product or service from the supplier or other party, as directed by the security criteria in the agreement.

AQ-5.4 Close the agreement in accordance with agreement security criteria.

References: [4] [86] [100] [111] [112] [113] [118]

K.2. Supply

The purpose of the *Supply* process is to provide an acquirer with a product or service that meets agreed requirements.

K.2.1. Security Purpose

- Provide an acquirer with a product or service that meets agreed security requirements.

K.2.2. Security Outcomes

- A response to the acquirer's request addresses the acquirer's security requirements.
- An agreement established between the acquirer and supplier includes security requirements.
- A product or service that satisfies the acquirer's security requirements is provided.
- Supplier security obligations defined in the agreement are satisfied.
- Responsibility for the acquired product or service, as directed by the agreement, is securely transferred.

K.2.3. Security Activities and Tasks

SP-1 PREPARE FOR THE SUPPLY

SP-1.1 Identify the security aspects of an acquirer's need for a product or service.

SP-1.2 Define the security aspects of the supply strategy.

Note: This strategy describes or references the security aspects of the life cycle model, risks and issues mitigation, and a schedule of security-relevant milestones. It also includes key security-relevant drivers and characteristics of the acquisition, such as responsibilities and liabilities, specific security-relevant models, security-relevant methods or processes, level of criticality, formality, and priority of relevant trade-off factors.

References: [4] [30] [61] [86] [98] [99] [100] [111] [112] [113]

SP-2 RESPOND TO A REQUEST FOR SUPPLY OF PRODUCTS OR SERVICES

SP-2.1 Evaluate a request for a product or service to determine the security-relevant feasibility and how to respond.

SP-2.2 Prepare a response that satisfies the security criteria in the solicitation.

References: [4] [30] [61] [86] [98] [99] [100] [111] [112] [113]

SP-3 ESTABLISH AND MAINTAIN AN AGREEMENT

SP-3.1 Negotiate and approve an agreement with the acquirer that includes security acceptance criteria.

Note 1: This includes configuration management, risk reporting, reporting of security measures, and security measure analysis; security requirements; secure development; security verification;

security validation; security acceptance procedures and criteria; regulatory body acceptance, authorization, and approval; procedures for transport, handling, delivery, and storage; security and privacy protections and restrictions on the use, dissemination, and destruction of data, information, and intellectual property; security-relevant exception-handling procedures and criteria; agreement change management procedures; and agreement termination procedures.

Note 2: The security aspects of the agreement also include applying all of the above to plans for subcontractor use.

SP-3.2 Identify necessary security-relevant changes to the agreement.

SP-3.3 Evaluate the security impact of necessary changes to the agreement.

Note: The basis for the agreement change may or may not be security-related. However, there may be a security-relevant impact regardless of the basis for the change. A security-relevant evaluation of the needed change identifies any security relevance and determines impact in terms of plans, schedule, cost, technical capability, quality, assurance, and trustworthiness.

SP-3.4 Update the security criteria in the agreement with the acquirer, as necessary.

References: [4] [30] [61] [86] [98] [99] [100] [111] [112] [113]

SP-4 EXECUTE THE AGREEMENT

SP-4.1 Execute the security aspects of the agreement according to established project plans.

Note: A supplier sometimes adopts or agrees to use acquirer processes, including security-relevant processes.

SP-4.2 Assess the execution of the security aspects of the agreement.

Note: This includes confirmation that all parties are meeting their security responsibilities according to the agreement.

References: [4] [86] [100] [111] [112] [113]

SP-5 DELIVER AND SUPPORT THE PRODUCT OR SERVICE

SP-5.1 Deliver the product or service in accordance with the agreement security criteria.

SP-5.2 Provide security assistance to the acquirer, per the agreement.

SP-5.3 Securely accept and acknowledge payment or other agreed consideration.

SP-5.4 Transfer the product or service to the acquirer or other party as directed by the security requirements in the agreement.

Note: This includes the transfer of hardware, software, and sensitive, proprietary, and classified information.

SP-5.5 Close the agreement in accordance with the agreement security criteria.

References: [4] [86] [100] [111] [112] [113] [118]

Appendix L. Change Log

L.1. Changes from NIST SP 800-160 Volume 1

This publication incorporates the following changes from the original edition (November 2016; updated March 21, 2018):

- Provides a renewed focus on the design principles and concepts for engineering trustworthy secure systems, distributing the content across several redesigned initial chapters
- Relocates the detailed system life cycle processes and security considerations to separate appendices for ease of use
- Streamlines the design principles for trustworthy secure systems by eliminating two previous design principle categories
- Includes a new introduction to the system life cycle processes and describes key relationships among those processes
- Clarifies key systems engineering and systems security engineering terminology
- Simplifies the structure of the system life cycle processes, activities, tasks, and references
- Provides additional references to international standards and technical guidance to better support the security aspects of the systems engineering process

L.2. Errata Update Summary

Table 6 shows changes incorporated into this publication. Errata updates can include corrections, clarifications, or other minor changes in the publication that are either *editorial* or *substantive* in nature. Any potential updates for this document that are not yet published in an errata update or a formal revision, including additional issues and potential corrections, will be posted as they are identified. See the publication details for this report.

The current release of this publication does not include any errata updates.

Table 6. Change Log

Publication ID	Date	Type of Edit	Change	Location