MTR210263
MITRE TECHNICAL REPORT

Principles for Trustworthy Design of Cyber-Physical Systems

Daryl Hild
Michael McEvilley
Mark Winstead

June 2021

Dept. No.: L266, L264
Project No.: 10AOH266-J1

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
13. SUPPLEMENTARY NOTES
14. ABSTRACT
15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER *(Include area code)*

Abstract

This report contains principles for systems engineering design of trustworthy cyber-physical systems (CPS), with emphasis on controlling the adverse effects (i.e., loss) that *might occur* as a direct or indirect result of the system delivering specified capability at specified levels of performance. Additionally, the document defines terms used in the definition, description, and interpretation of the principles.

The principles and terms are representative of the practices of the safety, security, survivability, and resilience communities and specialties – collectively the goals of these practices represent the "end objectives" the system must satisfy for trustworthy control of adverse effects. The concepts and theorems from the disciplines of computational science, control systems, systems engineering, software engineering, fault/failure tolerance, and mathematics – as employed collectively across the communities and specialties – constitute the "means to achieve" the end objectives.

The principles and terms are intended to be used as a starting point for discussion, vetting, employment, and ultimately acceptance by the engineering community. Their acceptance facilitates standardization within and across the general and specialty systems engineering practices and provides content suitable for inclusion in engineering bodies of knowledge.

The design principles are structured in three categories:

- Engineering design principles fundamental to managing complexity,
- Trustworthiness design principles fundamental to the design of a system for which there is justified confidence in its ability to function and produce outcomes only as intended, and
- Loss control design principles fundamental to achieving loss control objectives

The opportunity for future work based on these principles includes their extension and application to support systems engineering requirements, design, and analysis activities enabled by digital engineering environments, such as requirements and design patterns for the formal specification, verification, and modeling of the characteristics of specific types of cyber-physical systems and their capabilities, and the specification, verification, and modeling of adversity-driven loss scenarios.

Acknowledgments

The authors would like to express their appreciation to Ms. Melinda Reed, Director, Resilient Systems directorate, in the Office of Strategic Technology Protection & Exploitation (STP&E) and under the Office of the Under Secretary of Defense for Research and Engineering (OUSD(R&E)) [1], under whose direction the basis for this report was established as part of the ongoing collaboration across the government, the defense industrial base, and academia through the Resilient Systems directorate-hosted Cyber Resilient Weapon Systems (CRWS) workshop series.

The authors also acknowledge the contributions, comments, and perspectives offered by the participants in the CRWS workshops, the DoD Cyber Industrial Technical Advisory Group under the leadership of Rich Massey, The Boeing Company (chair) and William Shih, the Raytheon Company (co-chair), and The MITRE Corporation (Bill Bail, John Brtis, Rich Graubart, Frank Lynam, Beverly Ware).

[1] The Resilient Systems directorate, in the Office of Strategic Technology Protection & Exploitation (STP&E) and under the Office of the Under Secretary of Defense for Research and Engineering (OUSD(R&E)), has the responsibility for weapon systems engineering policy and for the standardization of weapon systems engineering practice in response to the challenges presented by contested cyberspace.

Table of Contents

List of Tables

This page intentionally left blank.

1 Introduction

This report contains principles for systems engineering design of trustworthy cyber-physical systems (CPS), with emphasis on controlling the adverse effects (i.e., loss) that *might occur* as a direct or indirect result of the system delivering specified capability at specified levels of performance.[2] Additionally, this report defines terms used in the definition, description, and interpretation of the principles.

The principles and terms are representative of the practices of the safety, security, survivability, and resilience communities and specialties – collectively the goals of these practices represent the "end objectives" the system must satisfy for trustworthy control of adverse effects. The concepts and theorems from the disciplines of computational science, control systems, systems engineering, software engineering, fault/failure tolerance, and mathematics – as employed collectively across the communities and specialties – constitute the "means to achieve" the end objectives.

The principles and terms are intended to be used as a starting point for discussion, vetting, employment, and ultimately acceptance by the engineering community. Their acceptance facilitates standardization within and across the general and specialty systems engineering practices and provides content suitable for inclusion in engineering bodies of knowledge.

1.1 Emphasis on Adverse Effects (Loss)

A multidisciplinary engineering perspective on the characteristics of cyber-physical systems[3] and on the adverse effects (loss) that might occur drives the statement and elaboration of the principles.

Adverse effects (loss) may occur as a result of a variety of intentional and unintentional causes, events, and conditions. These include the authorized or unauthorized use of the system, intentional acts of disruption or subversion, from faults and failures, and the by-product of emergence, side-effects, and feature interaction [27]. These adverse effects (loss) can be inconsequential to the business/mission objectives supported by the cyber-physical system, meaning that business/mission is achieved despite suffering an immediate or eventual adverse effect (loss) that could be prevented.

This potential for adverse effects (loss) translates into a need for *justified confidence* that the cyber-physical system can be used and sustained to ensure achievement of its intended purpose without suffering unacceptable adverse effects (loss). Justified confidence can be achieved with greater consistency, repeatability, and effectiveness if grounded in principled-based engineering trades and optimizations to enable delivery of system capability while limiting the extent of adverse effects (loss).

---

[2] This report is based on a report prepared for the Resilient Systems directorate, in the Office of Strategic Technology Protection & Exploitation (STP&E) and under the Office of the Under Secretary of Defense for Research and Engineering (OUSD(R&E)), as part of ongoing collaboration across the government, the defense industrial base, and academia through the Resilient Systems directorate-hosted Cyber Resilient Weapon Systems (CRWS) workshop series.

[3] The basis of this report was weapon systems. Weapon systems are a specific type of cyber-physical system that have the distinct purpose to deliver lethal force with the intent to cause harm when needed, where needed, and to the extent needed. The intent to cause harm differentiates weapon systems from most, if not all, other system types.

1.2 Goal of this Report

This report has the goal to enable discussion, vetting, and ultimately the acceptance of terms, concepts, and principles to advance the practice for trustworthy design of cyber-physical systems. There are numerous documents that contain definition and elaboration of principles, concepts, techniques, and terms beyond those used to produce this report. This report does not attempt to serve as an authoritative source, nor does it claim to be exhaustive and complete in coverage. Finally, this report recognizes that continuous change and advancement is necessary in response to variances in the sophistication and effectiveness of engineering methods, in the available technology, and in the adversity and adverse effects (loss) that systems to which systems are subjected.

1.2.1 Design Principles

The design principles are structured in three categories:

- Engineering design principles fundamental to managing complexity
- Trustworthiness design principles fundamental to the design of a system for which there is justified confidence in its ability to function and produce outcomes only as intended
- Loss control design principles fundamental to achieving loss control objectives

The following perspectives, insights, and assertions were used by the authors to inform the phrasing and elaboration of the principles included in this report:

- Cyber-physical systems operate in the physical domains of air (aircraft), land (surface vehicles), maritime (surface ships and submarines), space (spacecraft), and the virtual/information domain of cyberspace[4].
- Systems are a composed entity with both static and dynamic characteristics
- Engineering focus for the principles is driven by the cross-cutting objective to control adverse/loss effects
- The cyberspace context for the system and the cyber-physical characteristics of the system must be distinguished and addressed throughout the system life cycle
- Attacks, enabled or induced, by or within cyberspace, are a lens through which the system is viewed and assessed
- Engineering decision-making regarding design, trades, risk mitigation, and issue resolution must be based on their effectiveness to limit the types of losses typically associated with safety, security, survivability, and resilience objectives and concerns
- The actions of an intelligent adversary, while constituting a dominant driver of ubiquitous concern (i.e., independent of any single specialty), is one of numerous "adversity drivers" of concern, some of which can result in the same adverse effect/loss
- The engineering response to the intelligent adversary must consider not only what the design accomplishes, but the extent to which the design itself introduces adversity (i.e., *design adversity*). That is, while the design may be effective to solve one loss-based problem, it may create new loss-based problems

[4] Cyberspace is any interconnected digital environment of networks, services, systems, and processes (ISO/IEC 27102:2019). The U.S. DoD describes cyberspace as an operational domain within the information environment consisting of interdependent information technology infrastructures and their resident data (JP 3-12).

- Engineering seeks to optimize capability performance across competing needs and priorities. The safety, security, survivability, and resilience performance of the system must be explicitly factored into trade decisions for the optimization of capability performance
- There must be a balance in the extent to which engineering actions and decisions are threat-dependent (causal based) and threat-independent (effects based). Confidence in the quality of all data, to include threat data, must be verified before its use

1.2.2 Definitions

The definitions and elaboration provided in the glossary are intentionally extensive to capture the multidisciplinary perspective on same/similar concepts that happen to have different terminology, priority, perspective in their expression and use (e.g., susceptibility and vulnerability). The goal of the glossary is to inform consensus-building efforts and establishment of syntax and semantics necessary to state claims about the protection capability of cyber-physical systems, and to craft compelling, valid, and logical evidence-based arguments that stated claims are achieved with high confidence.[5]

1.3 Future Work

Engineering practice can benefit from standardized terms, concepts, theorems, principles, and methods for system design. This maturation aids in providing justified confidence in the effectiveness of system protections to enable delivery of system capability while limiting the extent of adverse effects (loss).

The opportunity for future work based on these principles includes their extension and application to support systems engineering requirements, design, and analysis activities enabled by digital engineering environments, such as requirements and design patterns for the formal specification, verification, and modeling of the characteristics of specific types of cyber-physical systems and their capabilities, and the specification, verification, and modeling of adversity-driven loss scenarios.

[5] The emphasis on Digital Engineering (DE) demands the specification of digital representations based on formal syntax and semantics. No significant progress can be made in the application of DE-based methods if the underpinnings of the digital representations are based on natural language and free-form or unstructured text.

2 Key Terms

Before discussing the design principles, we present the key terms that are used in the titles and elaborations.

To support the standardization of cyber-physical system engineering practice in response to the safety, security, survivability, and resilience concerns associated with contested cyberspace, each term includes reference(s) when used unmodified or with some adaptation, or the key reference(s) inspiring the definition. Elaborations of these key terms with notes of relevant insights are provided in the glossary. The definitions and elaborations are provided with emphasis to their use in support of cyber-physical system engineering.

Adversity	anything that can contribute to or result in a loss
Asset	an item (tangible or intangible) of particular value to stakeholders. [15][45]
Element	a discrete part of a system that can be implemented to fulfill specified requirements [14]; an identifiable part, one of the parts of a compound or complex whole [39].
Entity	a subject (active entity) or object (passive entity) with a distinct and independent existence that acts, accesses, is acted upon or is accessed.
Least	the smallest extent possible
Loss	The state of no longer having an asset, no longer having as much of an asset, having an undesired change in the condition of an asset, or having an undesired behavior of an asset.
Minimal	The smallest extent or amount practical
Minimize	reduce to the smallest extent or amount practical
Module	a well-defined 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. [25][39][15]
Resource	any usable part of a system that is controlled and assigned, or an item or asset that is used or consumed during execution of a function [14]
Rigor	quality of being detailed, careful, complete, and thorough
Susceptibility	the inability to avoid an adversity. [50][43]
Trust	(noun) belief that an entity meets expectations (verb) to believe that an entity meets expectations [[38], adapted]
Trustworthiness	well-founded assessment of the extent to which a given system, network, or component will satisfy its specified requirements, and particularly those requirements that are critical to an enterprise, mission, system, network, or other entity. [41]
Trustworthy	worthy of being trusted to satisfy the given expectations [25]
Vulnerability	the inability to withstand an adversity [50][43]

3 Loss Basis for Design

A trustworthy system must meet specified requirements reflecting stakeholder consensus across expectations, needs, priorities, concerns, and constraints. One specific concern is the ramifications associated with the loss of human and material assets. Protecting those assets against loss becomes a key dimension of trustworthiness, requiring the approach for the design of the system to have an explicit focus on controlling loss.

This section describes the asset classes that are the basis of cyber-physical system design and identifies the loss control objectives for cyber-physical system design to enforce constraints and to provide control to protect against loss.

3.1 Asset Classes to Define and Assess Loss

Loss concerns are based on the stakeholders' valuation of assets and their willingness to invest in protection of the asset throughout the life cycle of the asset.[6] In this document we focus on a generalization of all asset classes in terms of their loss concerns that span the scope of safety, security, survivability, and resilience.

- **System Capability Asset Class:** System capability is protected from loss if the system is able to deliver only the intended capability and produce only the intended outcomes at the specified level of performance[7]. Generally, the capability and performance are a function of (a) the role of the system to support achieving business/mission objectives and (b) the nature of the system (e.g., vehicular (aircraft, ship, train, truck, automobile), medical, financial, industrial, entertainment/recreational).
- **Human and Material Resource Asset Class:** Human and material resources are protected from loss if
    - Human resources are not injured, suffer illness, or killed;
    - Material resources are not destroyed or continue to function as needed when needed.
- **Technology Asset Class:** Technology is protected from loss if the technology is not exposed in an unauthorized manner (read/copy/stolen), or reverse-engineered in an unauthorized manner.
- **Data and Information Asset Class:** Data and information is protected from loss due to unauthorized alteration, exfiltration, infiltration, and destruction. This is the historical basis for computer security which over time evolved into cybersecurity. The data and information types include all forms of classified data and all forms of unclassified sensitive data:
    - Classified data and information are protected in accordance with an Executive Order, with the objective to protect sources, means, and methods.

[6] Assets have their own life cycle (creation, use, archive, destruction) that is independent of the life cycle of the cyber-physical system that may use or interact with the asset.

[7] "Intended" has two cases, both of which must be satisfied: (1) as intended per the design (design intent), and (2) as intended by the user (user intent). A system that delivers capability per the design intent but inconsistent with the user intent constitutes a loss. Example: the loss of control of a vehicle might result from failure in the control mechanism (failure to meet design intent) or through attack that takes control away from the user (failure to meet user intent for control)

- Unclassified but sensitive data and information protection includes critical program information, controlled unclassified information, controlled technical information, intellectual property, financial data and information, health and privacy data and information, and data that combined with other data constitutes sensitive information.

The system design must be optimized to achieve the differing protection needs of all four asset classes, to include addressing how those needs may vary, to include intentionally inhibiting protections, across all states and modes of the system.

There are additional non-tangible assets of significant importance to stakeholders. These include image, reputation, and trust. These non-tangible assets are affected, positively and negatively, by the success or failure to protect the 4 asset classes identified.[8]

3.2 Loss Control Objectives

The end state of protection is the preservation of the asset to the extent practicable, accepting there are limits to what can be done to reduce loss potential. Moreover, due to uncertainty, guaranteeing that loss does not occur is impossible, thus the need to address protecting against the effect of loss and any cascading of loss, i.e., the effect of loss is further loss and more unintended/undesired effects of loss.

Thus, protecting against loss and the unintended/undesired effects of loss holistically considers the full spectrum of possible loss across types of losses and loss effects associated with each asset class. This point is of particular importance considering that all forms of adversity are not knowable, and there is prudence in focusing on effect rather than cause when protecting against loss.

The loss control objectives (Table 1) address the realm of possibilities to control the potential for loss and the effects of loss given the limits of certainty, feasibility, and practicality. Collectively, these loss control objectives encompass the concerns of system safety, security, survivability, and resilience, and include all "cyber" interpretations of those objectives.

Table 1 – Loss Control Objectives

Loss Control Objective	Description	Elaboration
Loss Prevention	Prevent the loss from occurring	a. This is the case where a loss is totally avoided. That is, despite the presence of adversity: o the system continues to provide only the intended behavior and produces only the intended outcomes, o the desired properties of the system and assets used by the system are retained, o the assets continue to exist. b. This may be achieved by any combination of: o preventing or removing the event(s) that would cause the loss. The loss never occurs. o preventing or removing the condition(s) that allows the loss to occur. The loss never occurs. o not suffering an adverse effect despite the event(s) or condition(s). There are two cases: ▪ The adverse effect never occurs. The loss never occurs. ▪ The adverse effect occurs but is *delayed*, thereby making it inconsequential and having no significant effect. The loss occurs but is meaningless.

[8] As an example, a single significant failure or a trend of failures to adequately protect against loss for any one or combination of the 4 asset types can result in irreparable damage to an organizations image or reputation or result in a lack of trust in the organization.

Loss Control Objective	Description	Elaboration
		c. Terms such as *avoid, continue, delay, divert, eliminate, harden, prevent, redirect, remove, tolerate*[9]*, and withstand* are typically used to characterize approaches to achieve this objective such that a loss does not occur despite the system being subjected to adversity.
Loss Limitation	Limit the extent of the loss	a. This covers cases where a loss can or has occurred, and the extent of loss is to be limited. b. The extent of loss can be limited in terms of any combination of the following: o Limited dispersion (e.g., migration, propagation, spreading, ripple, domino, or cascading effects) o Limited duration (e.g., milliseconds, minutes, hours, days) o Limited capacity (e.g., diminished utility, delivery of function, service, or capability) o Limited volume (e.g., bytes of data, information) c. Decisions to limit the extent of loss may require prioritizing what constitutes acceptable loss across a set of losses, whereby the goal to limit the loss for one asset requires accepting a loss for some other asset. d. The extreme case of loss limitation is to avoid destruction of the asset. e. Terms such as tolerate, withstand, remove, continue, constrain, stop/halt, and restart fall into this category for the case where the loss occurs, and the system is able to, or enables the ability to, limit the effect of the loss. f. Loss recovery is one means of limiting loss. That is, the restoration of the asset, fully or partially, has the effect to limit dispersion, duration, capacity, or volume of the loss. o This is the case where action is taken by the system or where action is enabled by the system to recover (allow the recovery of) some or all of its ability to function (behave, interact, produce outcomes) and to recover assets used by the system (e.g., re-imaging, reloading or recreating information and data, including software in the system). g. Delay may also result in limiting loss when delay avoids the event until a time where the adverse effect is lessened, or when a delay enables a more robust response or quicker recovery. h. System and environmental conditions may be assumed to result in loss, but measures are taken to limit impacts. i. Terms such as *contain, recover, restore, reconstitute, reconfigure*, and *restart* are typically used to characterize approaches to achieve this objective.

[9] Tolerate refers to the objective of fault/failure tolerance, whereby adversity in the form of faults, errors, and failures are rendered inconsequential and do not alter or prevent realization of the intended behavior and outcomes of the systems. That is, the faults, errors, and failures are tolerated. As used in this report, 'tolerate' does not refer to a risk acceptance decision.

4 Design Principles Overview

Michael Watson defined systems engineering principles as "accepted truths that apply throughout the discipline of systems engineering, guiding the application of systems engineering" [26]. Adapting Watson's definition and criteria for general systems engineering principles to cyber-physical system design, we characterize the design principles as accepted truths about system design to achieve loss control objectives, with specific consideration and emphasis on the cyber-physical system and the adversity stemming from, but not exclusive to, the contested cyberspace environment. The design principles express the common aspects of, while not losing the nuances found in, principles used by the safety, security, resilience, and survivability disciplines.

The design for the system achieves the loss control objectives by controlling the adverse effects associated with the delivery of required capability. Control must be exercised over both the behaviors of the system and the outcomes produced by the system. For control to be effective, it must account for (a) achieving only the intended behaviors and outcomes, and (b) adversity in the form of the various cause-effect relationships that alter behaviors and outcomes and result in specific types of losses (i.e., adverse effects).

While the intended behaviors and outcomes to be exhibited by the system are known with confidence (they are expressed as system requirements), the adversity the system is subjected to (both adversity in the environment and internal adversity) and its effects are infinitely broad and are not known with confidence. An additional element of uncertainty is associated with the complexity of intentional adversity, which is compounded by adversity in the form of attacks by an intelligent, skilled, and motivated adversary.

The analysis that led to the design principles and their elaboration recognizes that a key objective for all of safety, security, survivability, and resilience is the reduction in uncertainty about the occurrence and effects of an adverse event. With that objective in mind, the common means to achieve the reduction is to eliminate to the extent possible, all hazards, susceptibility, and vulnerability. Where hazards, susceptibility, and vulnerability cannot be eliminated, it is then necessary to control their effects. The application of design principles is a part of the means to achieve both this elimination and the control.

We concluded that the following three categories sufficed to present the design principles.

- Engineering design principles that are fundamental to managing complexity
- Trustworthiness design principles that are fundamental to the design of a system for which there is justified confidence in its ability to function only as intended
- Loss control design principles that are fundamental to achieving the loss-control objectives

Our criteria for what constitute a principle follows. A principle must:

- Transcend cyber-physical system types
- Transcend context, including not being the application of another principle to a specific context
- Inform a broad, holistic perspective on achieving loss objectives
- Be descriptive (avoid being prescriptive)

- Be supported by literature and/or widely accepted in the profession (e.g., proven successful in practice across multiple organizations, multiple system types, and at least one loss discipline with clear broader applicability)
- Be focused, concise, and clear

The criterion of "supported by literature and/or widely accepted" is liberally interpreted. The authors believe all principles presented are universally accepted or well on a path to acceptance. Our liberal interpretation allows us to avoid distinguishing a principle from those concepts that may not be principles under more conservative interpretations.

4.1 Use of the Design Principles

Generally accepted systems engineering principles such as those captured by Watson [26] should be revisited for loss-driven considerations and clarified as needed [30].

Engineering judgement must be exercised in the application of the principles.

- Principles must not be applied as rules to be complied with, nor should they be prioritized, sequenced, or ordered for prescriptive application, or used individually or in groups as a basis to make judgements of conformance.
- Principles are subject to various priorities and constraints that may restrict or preclude their application. For example, the need for a small form factor or weight limits may restrict or prevent the usage of redundancy.
- Principles are subject to necessity and any associated priorities and constraints that stems from the system of interest context for their application. The system of interest that is the focus of an engineering effort may be a subsystem within a larger system or a system within a system of systems. Examples include:
    - The system of interest may be a part of a redundant path within a network system, reducing or eliminating any internal needs for redundancy.
    - The system of interest resides within a protected boundary, perimeter, or environment context that provides adequate protection to minimize detectability.
    - The system of interest relies on an enterprise service to provide some of its mediated access needs.
- Principles may be in tension with others, requiring judgements on which to prioritize for a given context. For example, the desire to reduce complexity conflicts with needs to use diversity, defense-in-depth, and other principles. Analyses and trades must inform the best approaches for application of the principles to optimize across design objectives.

As in all cases for engineering, the basis for design are the needs, concerns, priorities, and constraints of stakeholders.[10] Those needs, concerns, priorities, and constraints must explicitly include stakeholder concerns about loss (i.e., adverse effects) and adversity (i.e., causes and conditions that produce the loss), and any priorities and preferences to address loss and

[10] Some stakeholders have loss concerns that are not driven by business/mission objectives. For example, the acceptable protection of intellectual property (IP) is determined by the owner of the IP and not the users of the IP. Similarly, the protection of classified information is driven by loss concerns beyond the immediate users of the classified information.

adversity.[11] Such a loss basis for design ensures that design trade space decisions explicitly account for controlling loss due to adversity.

The significance of the kinds of losses associated with cyber-physical systems demands trustworthiness in the cyber-physical systems' ability to function only as intended despite all forms of adversity. The application of the principles serves to produce a design that results in trustworthy control over the system behaviors and outcomes, to optimize the system capability and protection against loss.

Finally, the design principles apply to the design for a new system; for the life cycle evolution of a system in the form of enhancement, modernization, technology refresh; and for the routine updates as well as upgrades made for performance enhancements, in response to hazards, susceptibility, vulnerability, or to specific attack methods. Expectations of life cycle evolution and future updates and upgrades may be a critical consideration impacting the application of design principles to the design of a new system.

4.2 Considerations for a Design Approach to Control Loss

MIL-STD 882E's *System Safety Design Order of Precedence* [5] inspires an optimized approach for system design for loss control. The safety design order of precedence reflects a design goal to eliminate *hazards* if possible. When a hazard cannot be eliminated, the risk associated with that hazard should be reduced to the lowest acceptable level within the constraints of cost, schedule, and performance by applying the system safety design order of precedence that identifies alternative design approaches and lists them in order of decreasing effectiveness.

The system safety design order of precedence can be applied to loss control with the following revision to the phrasing used in MIL-STD-882E. The revision shifts focus from "hazard" to "potential for loss". We take this approach because an analysis of how the term hazard might be interpreted for purposes other than system safety has not been accomplished so as to align with the aims of this report.[12] We substitute the phrase "susceptibility, hazard, and vulnerability" for the term "hazard" in the revision.

The safety design order of precedence is as follows:

1. **Eliminate the potential for loss through design selection.** Ideally, susceptibility, hazard, and vulnerability should be eliminated by selecting a design or material alternative that removes susceptibility, hazards, and vulnerability completely, and thus prevent loss.

   Example: The design selected for a *system function of interest* minimizes the number of interfaces to other systems (i.e., external interfaces) and the number of internal interfaces (i.e., interfaces with no interconnection to other systems). The minimization of interfaces (external and internal) is determined in consideration of the interface needs of *all system functions*, and results in an across-the-board optimization that does not overly constrain the design for the *system function of interest.* That is, the design results in less

[11] System requirements serve as the basis for engineering design. The explicit capture of stakeholders needs, concerns, preferences, priorities, and constraints in the system requirements ensures that all engineering activity will be conducted to account for the losses that are associated with the delivery of capability by the system.

[12] An interpretation of hazard in a security content can be found in [29].

susceptibility, hazard, and vulnerability than a design that incorporates additional and unnecessary internal and external interfaces.

Note that the design selection to control loss is accomplished to accommodate the need for mechanisms that provide mediated access, trusted communication, etc., as these engineered features and devices are necessary for a secure system.

2. **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 selection that would reduce the potential, severity, or extent of loss caused by the susceptibility, hazard, or vulnerability.

   Example: The selected design for the *system function of interest* has susceptibility, hazard, and vulnerability due to the system-level design trades made to satisfy the requirements for *all system functions*, due to emergence, and due to the limits of certainty. In response to these conditions, the design might consider functional domains, defense-in-depth layering, redundancy, and other approaches to further reduce susceptibility, hazard, and vulnerability.

   Note that the design alteration to control loss is accomplished to accommodate the need for mechanisms that provide mediated access, trusted communication, etc., as these engineered features and devices are necessary for a secure system.

3. **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, or extent of loss.

   Example:

   There are two general types of engineered features and devices that are employed to address the potential for loss associated with the *system function of interest.*

   - **Mandatory security features and devices**: Mandatory security features and devices are those that apply foundational security principles for the interfaces. As an example, each interface must have mediated access to control access to, and the use of, 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 elements can supplement the design selection to achieve the required protection.

4. **Provide warning systems and devices.** If design alteration, material selection, and engineered features and devices are not feasible or do not adequately lower the potential, severity, or extent of loss caused by the susceptibility, hazard, or vulnerability, employ engineered detection and warning systems and devices to alert personnel 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.

Example: Engineered anomaly detection features can be used to provide situation awareness data and warnings to system users.

5. **Incorporate signage, procedures, training, 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 susceptibility, hazard, or vulnerability, incorporate procedures, training, signage, and proper protective equipment. Procedures and training should include appropriate warnings and cautions and may prescribe the use of personal protective equipment (PPE). For critical losses, 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 materials address proper use of the *system function of interest* as well as the use of mediated access functions, redundant capabilities, and warning systems, to include all relevant cautions and warnings.

5 Engineering Design Principles

The engineering design principles (Table 2) provided in this section are to manage complexity and otherwise aid in understanding the engineered system in depth. They have critical importance to achieve loss control objectives given the complexity in understanding loss in context (based on how the system is intended to be utilized and sustained).

Complexity increases analysis workloads and reduces confidence in that analysis. Complexity increases costs and difficulty in performing systems analyses for loss; conceivably systems may be too complex to be analyzed for adequate assurance [48].

Table 2 – Engineering Design Principles

Title	Description
Clear Representation	The abstractions used to characterize the system for systems engineering purposes should be simple, well-defined, accurate, precise, necessary, and sufficient.
Composition Principles	System complexity should be managed through the structured decomposition of the system into cohesive constituent elements, and the structured composition of the constituent elements to deliver required capability.
Reduced Complexity	The system design should be as simple as practicable.

5.1 Clear Representation

The abstractions used to characterize the system for systems engineering purposes should be simple, well-defined, accurate, precise, necessary, and sufficient.

Elaboration:

Abstraction [15] is one aid in managing complexity. Clarity in the abstract representations of the systems facilitates accurate understanding of the system and how it functions to deliver required capability. Clarity in the abstract representations of the system reduce the potential for misunderstanding or misinterpretation of what is represented by the abstraction. It is therefore important to ensure that the appropriate rigor is applied in the creation of system abstractions during design. Specifically, clarity in the abstract representation of the system requires using well-defined syntax and semantics, with elaboration as necessary to ensure the representations are well-defined, precise, necessary, and sufficient.

Clear representation promotes confidence in analysis and verification, as well as the correct use of the system.

Note: a common form of abstraction is models, including SysML-based models.

Also called: Clear Abstraction

Ref: [25][41][49][15]

5.2 Composition Principles

System complexity should be managed through the structured decomposition of the system into cohesive constituent elements, and the structured composition of the constituent elements to deliver required capability.

Elaboration:

Watson's Systems Engineering principles [26] includes "A systems engineering focus is a progressively deeper understanding of the interactions, sensitivities, and behaviors of the system, stakeholder needs, and its operational environment". For managing system's complexity for understanding in terms of the loss control objectives, this requires composing the system with modularity, layering, and partially ordered dependency relationships to achieve intended, and only intended, behaviors and outcomes.

Modularity is the system design technique to 'divide and conquer' – sub-divide the system into smaller well-defined cohesive components and assemblies, generically referred to as modules. Modular design is based on functions and data, and may 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 [41].

Partially ordered dependencies among modules (i.e., if "module a" depends on "module b", "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, if circular dependencies are constrained to occur within layers and minimized within each layer, the inherent problems of circularity can be more easily managed. Partially ordered dependencies also facilitate system testing and analysis, and enables 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, and relationships, and how the system delivers the required capability. The relevant principles are used for decomposition to produce a design capable of satisfying the system performance requirements, to include the requirements that reflect the loss control objectives.

The structured composition of the constituent elements must adhere to the principle of *Compositional Trustworthiness* to provide a legitimate basis to support claims about how the system composes based on the application of modularity, layering, and partially ordered dependencies to achieve intended, and only intended, behaviors and outcomes.

Ref: [17][25][41][2][42]

5.3 Reduced Complexity

The system design should be as simple as practicable.

Elaboration:

Many engineered systems are complex both in terms of their internal structure, including internal interactions, and their interactions with their environment. Some levels of complexity are inherent, unavoidable, and must be accepted; the objective is to ensure the design reflects the extent to which complexity can be reasonably minimized (i.e., avoid unnecessary complexity). Simplicity in design reduces complexity and allows increased confidence in the ability to understand the system design. A simpler design is less prone to erroneous interpretation for implementation, for system analysis, and for system verification [33]. The reduced complexity contributes to confidence in technical understanding of the design, enabling better informed trade decisions.

Complexity lies in the system structure, interfaces, dependencies, and data and control flows. Complexity is impacted by how the system is decomposed into constituent elements, aggregates of elements (e.g., subsystem, assemblies), and the composition of those elements to comprise the system. Identifying and assessing loss scenarios, susceptibilities, and vulnerabilities is made more difficult by complexity; thus, reducing complexity facilitates the identification and assessment of loss scenarios, hazards, susceptibility, and vulnerability to all forms of adversity.

Also called: Simplification, Economy of mechanism, Adopt sweeping simplifications

Ref: [17][41][47][16][33]

6 Trustworthiness Design Principles

Trustworthiness-related principles (Table 3) are based on the historical meaning of trust and trustworthiness and their use as the basis for the design of secure systems [41][49][39][6].

We summarize the meaning of trust and trustworthiness as follows [41]. A full discussion is provided in the Glossary:

- Trustworthiness: the demonstrated worthiness of an entity to be trusted. Trustworthiness, being demonstrated, is therefore based on evidence that supports a claim or judgement of being trustworthy, that is, worthy to be trusted.
- Trust: a belief that an entity can be trusted. "Can" implies that trust may be granted to an entity whether the entity is trustworthy or not.

In this report, the focus is on trustworthiness. Trustworthiness is a cross-cutting objective in the design of cyber-physical systems due to the high consequences of the failure to behave and to produce outcomes only as intended. The terms *trust* and *trusted* are used to mean "the decision is made to trust because the required trustworthiness is demonstrated".

Realistically, not every system element may have the trustworthiness reflecting the most adverse effect associated with its failure. Consequently, trustworthy-related principles facilitate necessary design trade space decisions (See Commensurate Rigor and Commensurate Trustworthiness). Further, any claim of trustworthiness should be substantiated, must account for trustworthiness in terms of the compositional trustworthiness of an aggregation of system elements, and reflect properties of trustworthy system control (Section 6.8). Finally, the design of the system should seek to maximize the trustworthiness of individual elements and the trustworthiness of aggregates of elements that must be trusted, while minimizing the number of elements that must be trusted.

Table 3 – Trustworthiness Design Principles

Title	Description
Commensurate Rigor	The rigor associated with the conduct of an engineering activity should provide the confidence required to address the most significant adverse effect that can occur.
Commensurate Trustworthiness	An element must be trustworthy to a level commensurate with the most significant adverse effect that results from a failure of that element.
Compositional Trustworthiness	The design for the system should be trustworthy for each aggregate composition of interacting elements.
Hierarchical Protection	An element need not be protected from more trustworthy elements.
Minimized Trusted Elements	A system should have as few trusted elements as practicable.
Self-Reliant Trustworthiness	The trustworthiness of an element should be achieved with minimal dependence on other elements.
Substantiated Trustworthiness	System trustworthiness judgements must be based on evidence, demonstrating that the criteria for trustworthiness have been achieved.
Trustworthy System Control	The design for system control functions conforms to the properties of the generalized reference monitor.

6.1 Commensurate Rigor

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

Elaboration:

Commensurate rigor ensures 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; in method and tool selection; and in personnel selection.

The more severe an adverse effect is, the more confidence is needed in achieving loss control objectives. Rigor is a key means to provide confidence in the results of a completed activity. Generally, an increase in rigor translates into an increase in confidence in the results of the activity. Further, increased confidence reduces uncertainty that also can 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 practice [14].

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. The increased cost should be justified by acquiring confidence about system performance to limit loss while also addressing the system's ability to deliver capability. Therefore, the rigor associated with an engineering activity should be commensurate to the significance of the most adverse effect associated with the activity.

Circumstances, such as achieving a needed certification, e.g., the Navy's SUBSAFE certification, may have a commensurate rigor criteria or requirement, implicitly or explicitly [32]. The achievability of many requirements within such criteria may be impacted, especially the achievability within cost and schedule constraints. The outcome is a trade among cost, schedule, and performance.

Ref: [14][41][32]

6.2 Commensurate Trustworthiness

An element must be trustworthy to a level commensurate with the most significant adverse effect that results from a failure of that element.

Elaboration:

A trusted element exhibits properties of trust continuously for the duration of the time that it is depended upon by other system elements. The amount 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 would allow (a) expressing the trust that is to be placed in an element, (b) expressing the trustworthiness that is exhibited by an element, and (c) comparing the trustworthiness of different elements. This principle is particularly relevant when considering systems and elements in which there are complex chains of trust dependencies.

Ref: [25][41]

6.3 Compositional Trustworthiness

The design for the system should be trustworthy for each aggregate composition of interacting elements.

Elaboration:

The trustworthiness of an aggregate of composed elements cannot be assumed based on the trustworthiness assertions of each element in the aggregate. Further, the trustworthiness of an aggregate of composed trustworthy components 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 elements. Each system function results from the emergent behavior of a composed set of elements. Likewise, the trustworthiness of a composed set of elements is an emergent property of the composition. Therefore, the trustworthiness of the composed set of elements (aggregate) for a given system function must be determined by treating the aggregate as a single discrete element.

The compositional trustworthiness principle addresses how one is able to make an argument 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.

Ref: [41][14][32][40]

6.4 Hierarchical Protection

An element need not be protected from more trustworthy elements.

Elaboration:

Hierarchical Protection is a simplifying assumption for trade decisions to determine where emphasis is placed in providing protection and the extent of the protection effectiveness.

The simplifying assumption of hierarchical protection introduces susceptibilities to elements dependent on the more trustworthy element. This simplifying assumption relies on (1) validated trust assertions about the more trustworthy element and (2) acceptable uncertainty associated with behavior outside the scope of the validated trust assertions. For example, systems may include a human element and often the human element is 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 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 all the less trustworthy applications it supports, but the applications, do not need to protect themselves from the operating system kernel.

Ref: [41][44]

6.5 Minimized Trusted Elements

A system should have as few trusted elements as practicable.

Elaboration:

Minimizing trusted 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 an 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 elements given the functions they provide, and then how those elements are best distributed throughout the architecture, where there is justified need for the distribution. Minimization of trusted elements is one of several considerations in making that decision.

Trusted elements are generally costlier to construct, owing to increased rigor in engineering processes. They also require more analysis to qualify their trustworthiness.

Minimizing the number of trusted elements reduces the cost of analysis (decreases the size, scope, and complexity of analysis). When the minimization of trusted elements considers the principle of commensurate protection, the cost-effectiveness of the analysis is also ensured (cost of the analysis is justified by the extent of trust required).

Historically, the analysis of interactions between trusted elements and untrusted system elements is one of the most important aspects of the trust-based verification of system security performance.

Also referred to: Minimized Security Elements, minimization of what must be trustworthy

Ref: [25][41][44][18]

6.6 Self-Reliant Trustworthiness

The trustworthiness of an element should be achieved with minimal dependence on other elements.

Elaboration:

The ideal case of an element's trustworthiness occurs when the trustworthiness claim is not dependent on protection from another element.

When an element is dependent on some other element 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 an element exhibits self-reliant trustworthiness includes:

- The trustworthiness objective for the capability
- The trustworthiness of the element in providing the capability
- The extent to which the capability provided by an element is dependent on another element
- The extent to which the trustworthiness associated with a capability is dependent on another element

Self-reliance is directly related to the notion of "self", meaning, an argument for self-reliant trustworthiness can be applied at the discrete 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 therefore dependent on, some other entity). Likewise, when a capability is distributed across multiple elements, self-reliant trustworthiness requires that the trust expectations for the capability are properly allocated across the elements comprising the distributed capability.

The judgement that an 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 required, or in the assumptions associated with the requirement requires reassessment to determine that the element remains self-reliantly trustworthy.

Self-reliant trustworthiness is tied to anomaly detection; requiring the self-reliant element to have self-reliant anomaly detection that is complete in coverage of all its trust assertions.

Self-reliant trustworthiness differs from self-reliant operation. The Apollo Lunar Module was self-reliant trustworthy to complete its mission and protect life during the mission, but operationally dependent on rocket boosters and other elements to place it in the environment where it was able to then perform its mission. NSA-approved Type-1 cryptographic modules are examples of technical solutions exhibiting self-reliant trustworthiness with high confidence for all design requirements. However, a Type-1 cryptographic module is operationally reliant on other means for the generation of keys and receiving timing information necessary to perform its design function in a trustworthy manner.

Ref: [41]

6.7 Substantiated Trustworthiness

System trustworthiness judgements must be based on evidence, demonstrating that the criteria for trustworthiness have been achieved.

Elaboration:

Trustworthiness should not be assumed, but rather substantiated through evidence that clearly enables determination of the extent to which an entity is worth being trusted. This helps 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: "elements are assumed to be guilty until proven innocent".[13] The approach is characterized by a design mentality where all components involved in the design context (an element and the elements with which it interacts) are treated with a mutually suspicious mindset [25][41]. Such mutual suspicion reflects cautious distrust; the feeling or thought that something not desired, not wanted, or not expected is possible or can happen. The design for every system element should reflect a lack of trust in interacting elements or to even itself. This suspicion assumes *element non-performance* and addresses the following two cases:

[13] 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).

1. Interacting element suspicion (mutual suspicion): The design for the element-of-interest is based on the non-performance of elements it interacts with and how their non-performance can influence the behavior and outcomes produced by the element-of-interest. Mutual suspicion may also be referred to as zero trust.[14,15] Designing to mutual suspicion is reinforced by applying 'least privilege' to all entities (so an element executes with only the privileges needed, mitigating harm that may be created) while applying least persistence so each element is minimally exposed.
2. Self-suspicion: The design for the 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 initiation of the element.

This approach forces the designer to assume things will not go right, and to rigorously seek evidence that demonstrates the effectiveness of the design when things do go wrong.

Considerations for element non-performance include:

- Expectation that design elements will behave and produce outcomes that are inconsistent with its design intent.
- Constraints, assumptions, and preconditions associated with achieving threshold performance.
- Intentional and unintentional events and conditions, and may be referred to by terms like fault, error, failure, and compromise.

Ref: [41][49][28]

6.8 Trustworthy System Control

The design for system control functions conforms to the properties of the generalized reference monitor.

Elaboration:

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.

The reference monitor concept 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 [6]. As a concept for an abstract machine, it does not address any specific implementation. A reference validation mechanism, a combination of hardware and software, realizes the reference monitor concept to provide the access mediation foundation for a secure system [20].

The reference monitor concept has three requirements providing design assurance of its realization as a reference validation mechanism:

[14] Zero trust means only that an entity is not trusted; zero trust does not mean that the entity is not trustworthy.

[15] Zero trust is not to be confused with Zero-Trust Architecture (ZTA).

- The reference validation mechanism must be tamper-proof, ensuring that its integrity and validity is 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. This has the purpose to ascertain that the reference validation mechanism works correctly in all cases.

For trustworthy system control, an addition of a fourth attribute is made with the resulting collective being referred to as "NEAT". NEAT is a description of the four necessary attributes of security protection mechanisms [9], i.e., being **n**on-bypass-able, **e**valuate-able, **a**lways invoked, and **t**amperproof. Successful achievement of the attributes will prevent interference of outside entities on a protection mechanism, or controller. More specifically:

- A protection mechanism or feature should not be circumventable, or non-bypass-able.
- The mechanism or feature should be evaluate-able: sufficiently small and simple enough to be assessed to produce adequate confidence in the protection provided, constraint (or control objective) enforced, and mechanism implemented correctly. The assessment includes the analysis and testing needed.
- If a mechanism or feature is not always invoked, then the protection will not be continuous.
- Finally, a protection mechanism or feature must be tamperproof: the protection functions nor the data the functions depend upon cannot be modified without authorization.

Additionally, trustworthy system control uses protective control: protective control encompasses control, safety, and security concepts to establish a control system capability that sufficiently:

- Enforces constraints to achieve only the intended system behaviors and outcomes,
- Provides self-protection against targeted attack on the control system, and
- Is absent of self-induced emergent, erroneous, unsafe, and non-secure control actions.

The notion of "protective control" underlies the loss control objectives and transforms the approach for design to not be dependent on having detailed knowledge of the capability, means, and methods of an attacker. 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 attacker.

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.

Ref: [49][20][6][9]

7 Loss Control Design Principles

The loss control principles (Table 4) are intended to produce a design that results in *trustworthy control* over the system behavior and outcomes, to deliver the required system capability and to protect against loss.

Table 4 – Loss Control Design Principles

Title	Description
Anomaly Detection	Any salient anomaly in the system or in its environment is detected in a timely manner that enables effective response action.
Commensurate Protection	The strength and type of protection provided to an element must be commensurate with the most significant adverse effect that results from a failure of that element.
Commensurate Response	The design should match the aggressiveness of an engineered response action's effect to the needed immediacy to control the effects of each loss scenario.
Continuous Protection	The protection provided for an element must be effective and uninterrupted during the time that the protection is required.
Defense-in-depth	Loss is prevented or minimized by employing multiple coordinated mechanisms.
Distributed Privilege	Multiple authorized entities must act in a coordinated manner before an operation on the system is allowed to occur.
Diversity (Dynamicity)	The design delivers the required capability through structural, behavioral, or data or control flow variation.
Domain Separation	Domains with distinctly different protection needs should be physically or logically separated.
Least Functionality	Each element should have the capability to accomplish its required functions, but no more.
Least Persistence	System elements and other resources should be available, accessible, and able to fulfill their design intent only for the time they are needed.
Least Privilege	Each element should be allocated privileges that are necessary to accomplish its specified functions, but no more.
Least Sharing	System resources should be shared among system elements only when necessary, and among as few system elements as possible.
Loss Margins	The system is designed to operate in a state space sufficiently distanced below the threshold at which loss occurs.
Mediated Access	All access to and operations on system elements are mediated.
Protective Defaults	The default configuration of the system provides maximum protection effectiveness.
Protective Failure	A failure of a system element should neither result in an unacceptable loss, nor invoke another loss scenario.
Protective Recovery	The recovery of a system element should not result in, nor lead to, unacceptable loss.
Redundancy	The design delivers the required capability by the replication of functions or elements.

7.1 Anomaly Detection

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

Elaboration:

The purpose of anomaly detection is to identify the need to take corrective action to address a loss condition that has occurred or to address a loss condition that will occur if conditions affecting the system behavior are allowed to persist. Anomaly detection is critical to achieving the loss control objectives to prevent and limit loss and its adverse effect.

The detection of such anomalies requires monitoring system behaviors and outcomes to confirm that they have not deviated from the design intent, and monitoring conditions in the environment to identify or forecast those 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 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 judgement of appropriate response.

Anomaly detection can be implemented at varying levels of abstraction (system, sub-system, assembly, function, mechanism, etc.), and may occur in periodic, aperiodic, or event-driven manners.

The basis for anomaly detection within the system is the expectation that system behaviors, interactions, and the outcomes 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 integrity, correctness, and trustworthiness of system elements; results of system behavior; state consistency; continuity of function; system configuration; and the abuse or misuse of the system.

The basis for anomaly detection in the environment differs from that of the system because the environment is not within the control of the system. The environment presents a 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, requiring a planned response to prepare for an impending difficulty or crisis (e.g., "batten down the hatches").

Anomaly detection requires capturing data to support all intended response actions for a detected anomaly, including attribution-related data. Consequently, the rigor in data describing the anomaly must be commensurate with the consequences of the loss scenarios associated with the anomaly as well as the consequence of wrong responses to address the detected anomaly. Response 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 element while an

undesired condition persists, or wrongful termination of an individual. Rigor in attribution 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 against which actual behavior and outcomes can be compared to enable conclusions, with confidence, that an anomaly exists or to determine or forecast that an anomaly is about to occur. System models includes 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, self-test, built-in test, monitoring, logging, and auditing.
- Data Interpretation: 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. Within safety, poorly designed lines of defense for defense-in-depth have been found to conceal emerging dangerous system states and conditions, especially to human observers [16]. The system design must minimize the difference between estimated system states and conditions and actual system state and condition.

There are two approaches to anomaly detection:

- Self-anomaly detection: An entity has no dependency on another entity to detect an anomaly within the scope of its intended design intent. Self-anomaly detection usually involves an axiomatic or environmentally enforced assumption about its integrity. Trusted elements typically have the capability for self-anomaly detection. 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, and 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.

Also see in the literature: self-analysis, accountability and traceability, comprehensive accountability, recording of compromises, real time analysis, anomaly/misuse detection, observability-in-depth.

Ref: [25][44][16]

7.2 Commensurate Protection

The strength and type of protection provided to an element must be commensurate with the most significant adverse effect that results from a failure of that element.

Elaboration:

The strength and effectiveness of the protection for an 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 element or the trust that is placed in the element.

The protection can come in the form of the 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 vs. centralized design, a concept sometimes referred to as secure distributed composition [41].

Furthermore, the confidence in the effectiveness of the protections provided to an element should also increase commensurate to the need. This is addressed by Commensurate Rigor.

Ref: [41] [49] as Inverse Modification Threshold

7.3 Commensurate Response

The design should match the aggressiveness of an engineered response action's effect to the needed immediacy to control the effects of each loss scenario.

Elaboration:

The selected response to a detected anomaly should be based on consideration of three factors to determine the effect that the response has on the loss and on the system:

1. The expected effectiveness and aggressiveness of the response to directly address the anomaly and to prevent or limit the loss
2. The direct, residual, or side-effect of the response on the system
3. The opportunities that remain to take some other response action should the selected response fail to achieve the intended result.

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

A commensurate response requires consideration of the response-effect-consequence relationship associated with a specific loss. Ideally, for any given need for 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 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 typically dictates 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 most impactful action possible to prevent the loss from continuing or escalating and does so without consideration of the possible 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 possible loss, the option for a graduated response may not exist. Commensurate response is aided by early detection, which increases graduated response options.

Ref: [16]

7.4 Continuous Protection

The protection provided for an element must be effective and uninterrupted during the time that the protection is required.

Elaboration:

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 required capability while controlling loss.

Continuous protection requires adherence to the following principles:

- The principle of trustworthy system control (i.e., every controlled action is constrained by the mechanism, the mechanism is able to protect itself from tampering, sufficient assurance of the correctness and completeness of the mechanism can be ascertained from analysis and testing); and,
- The principles of protective failure and protective recovery (i.e., preservation of a protective state during error, fault, failure, and successful attack; preservation of a protective state during 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, and to the transitions between those configurations, states, and modes.

The design for the system must ensure that protections are coordinated and composed in a non-conflicting and mutually supportive manner across the non-behavioral aspects of system structure and the behavioral aspects of system function and data flow.

While the design for continuous protection applies for the entirety of the time that the protection is required, there may be cases where, by design, protection capability is intentionally disabled (e.g., Battleshort,[16] 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

[16] Battleshort is a switch used to bypass normal interlocks in mission critical equipment (e.g., equipment which must not be shut down or the mission function will fail) during battle conditions [7].

protection applies only for the entirety of time that the protection is required and not knowingly and intentionally disabled.[17]

Also referred to as: Continuous Protection of Information.

Ref: [49]

7.5 Defense-in-depth

Loss is prevented or minimized by employing multiple coordinated mechanisms.

Elaboration

Coordinated deployment of depth in protections for a system helps avoid single points of failure in protection. Defense-in-depth has several pillars:

- Multiple lines of defenses or barriers should be placed along loss scenario sequences;
- Loss control should not rely on a single defensive element (hence the “depth” qualifier in defense-in-depth); and
- The successive barriers should be diverse in nature and include technical, operational, and organizational barriers.

Defense in depth requires the employment of coordinated mechanisms (active) within an architectural structure (passive) that achieves the "depth" characteristic.[18] Ideally, the initial lines of defense prevent loss, while subsequent lines of defense (when needed) block loss scenario escalation and/or contain loss and potential consequences. A defense-in-depth strategy examines loss scenarios for those points of opportunity to prevent or contain loss and leverages many or all opportunities to use active or passive mechanisms or constraints to meet loss control objectives.

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

While defense-in-depth distributes 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 component is a potential single point of failure, a possible bottle neck to system performance, or raise other concerns; distributed composition of a defense layer may provide additional options within the coordination of layers.

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

Also referred to as: layered defense

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

[18] While the elaboration is limited to the machine, defense-in-depth may involve the combination of technical, operational, and organizational elements.

Ref: [41][49][47][16]

7.6 Distributed Privilege

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

Elaboration

Distributed privilege is a means to prevent a single authorized entity from performing an erroneous action, regardless if that action is performed with or without the intent to do so.

Distributed privilege requires that an erroneous action can only be performed if the multiple entities agree to cooperate to do so, for either legitimate (e.g., override of the protection in extreme cases) or illegitimate purpose (e.g., collusion to intentionally take improper action). For the case of attacks on an operation, distributed privilege forces the attack to target all the entities to which privilege is distributed.

Distributed privilege separates, divides, or in some other manner distributes the privilege required to perform an operation among multiple entities. The distribution of privilege includes a set of rules, conditions, and constraints that describe how the multiple entities must interact through positive action before a requested operation can proceed and complete. The rules, conditions, and constraints may reflect combinations of the following, all of which require that multiple conditions be met for the operation to proceed:

- 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 (e.g., a transaction with interlocks whereby the transaction only succeeds if each step in the transaction completes and in doing so, satisfies the pre-conditions for the next step in the transaction); and
- Parallel actions: Multiple entities execute sequences concurrently, whereby success is either a consensus of the results of each concurrent action or a voting among the participates.

See also in literature: separation of privilege, separation of duty

Ref: [17][49]

7.7 Diversity (Dynamicity)

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

Elaboration:

A design incorporating diversity helps to avoid common mode failures and introduces unpredictability to an adversary; complicating their planning and execution of where, when, and how to target their attacks. While the design’s behaviors may be unpredictable from the

viewpoint of the adversary, the design itself must be predictable and verifiable in achieving only the intended outcomes.

Diversity options include variety in the system structural and architectural design elements, variety in the system functional and behavioral elements, variety in the interfaces and interconnections between interfaces, variety in data and control flow, and variety in 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)
- 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, and the ability to change aspects of the system behavior, structure, data, or configuration in a random but nonetheless verifiable manner)

An example of diversity is the U.S. strategic communications, which leverages a variety of communications paths (e.g., HF, SATCOM, wired lines, fiber optic lines). The outcome is the ability to assure strategic messages are delivered despite a variety of adversities, such as atmospheric nuclear blasts that interfere with RF communications or space weather events known to interfere with ground to satellite communications.

Any 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 due to 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 intended functional behavior and produce only the intended outcomes and does so in a manner that allows for control over side-effects, emergence, and feature interaction.

Diversity may have a cost (hardware, software, maintenance, training, assurance) greater than the value or effectiveness it provides.

Diversity options include intentionally designed regular or irregular changes in the system, e.g., using dynamicity. This results in unpredictability and uncertainty to the adversary, complicating their planning of effective attacks, and can provide required performance despite other adversity. Dynamic change may refer to either (a) shifting the target or (b) shifting the behaviors of a target in performing its purpose.

An example of dynamicity is frequency hopping with wireless communications, which complicates both interception of signal and jamming of signal.

A design incorporating dynamicity can serve several purposes: (1) it complicates the attack planning of an intelligent adversary, (2) it reduces the potential for non-adversarial adversity to have an effect on the system, (3) it provides capability and margin to deliver required capability while reducing actual losses, and (4) it protects against the effects of an attack.

The uncertainty and diminished predictability associated with the use 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 intended. It is important to differentiate where the uncertainty lies: (a) uncertainty in how the system achieves an end objective (i.e., the means to an end), or (b) 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 space that is specific to diversity-based and dynamicity-based designs.

Also referred to: Reorganization

Ref: [25][47][33]

7.8 Domain Separation

Domains with distinctly different protection needs should be physically or logically separated.

Elaboration:

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 other domain, enhancing the protection of a domain. This is achieved by control of data and information flow between domains as well as control over the use of 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 (susceptibility) and protecting external entities from erroneous behavior that occurs within the domain (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 separation of roles or privilege (least privilege). For example, a computer may separate an ‘administrative’ or ‘supervisor’ domain, distinct from user domain(s). This administrative domain is accessible only by system administrators and distinctly administrative functions may only be executed from the administrative domain by administrators. Similarly, data intended to only be accessed by administrators and administrative functions, e.g., system configurations, is stored and accessed only within that domain, aiding enforcement of needed protection of the data.

Domain separation commonly 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 implementing 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 with the environment to deliver capability. Every interface that results from design decisions can diminish domain separation while achieving the requirements for system capability. External requests for resources or functions within the protected subsystem are arbitrated at these interfaces. Firewall, data diodes, and cross domain solutions (CDS) are examples of mechanisms enabling 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 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.

Ref: [44][49]

7.9 Least Functionality

Each element should have the capability to accomplish its required functions, but no more.

Elaboration:

Susceptibility and vulnerability increase unnecessarily when an element provides more functionality than is needed to achieve its intended purpose. Least functionality reduces the potential for susceptibility and vulnerability and also reduces the scope of analysis of the element's trustworthiness and loss potential.

The most aggressive interpretation of least functionality is to completely avoid including any element functions that are not required. Where that is not possible or practical, the unnecessary functions of the element should at best be fully disabled, disarmed, or put into a "safe" mode that prevents the functions to be used. In all other cases, mediated access can be used to prevent all 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 that contain functions beyond those required to fulfill its purpose. In such cases, the components should be configured to enable only the functions that are required to fulfill its purpose and to prohibit or restrict functions that are not required to fulfill its purpose.

Least functionality is reflected in concepts such as removal of dead code, unreachable code, and unnecessary functionality.

Ref: [41][49]

7.10 Least Persistence

System elements and other resources should be available, accessible, and able to fulfill their design intent only for the time they are needed.

Elaboration:

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 elements and resources until needed, and to completely remove elements or remove power from elements and resources when they are no longer required. Where that is not possible or practical, those elements and resources should at best be fully disabled, disarmed, or put into "safe" mode to prevent their ability to function or to be used. At a minimum, mediated access should include constraints on the time and duration of their use.

Three conditions must be satisfied for an active element or resource to be usable, with two of these conditions applying to non-active elements or resources:

- Presence (active and non-active): The element or resource must be installed, loaded, residing in memory, configured, etc.
- Accessible (active and non-active): The element or resource can be invoked, interacted with, or operated on.
- Able to function (active): The element or resource must be able to execute, powered on, enabled, armed, etc., to deliver a service or perform a function

Least persistence is reflected in concepts such as sanitizing, erasing, clearing memory and storage locations; disabling, removing, disconnecting network ports, system interfaces, and the services provided by system interfaces; powering off and unplugging hardware when not needed; instantiating software just before needed and de-instantiating after it is no longer needed.

Least persistence has added benefits that include simplifying the processes of:

- Cleansing of the system element to remove corrupted aspects or side effects;
- Re-establishing the system element to a known state (i.e., a refresh); and
- Minimizing the period of time that 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.

Also referred to as: “non-persistence”

Ref: [46]

7.11 Least Privilege

Each element should be allocated privileges that are necessary to accomplish its specified functions, but no more.

Elaboration:

Least privilege is a pervasive principle of secure system design to contain damage and to simplify analysis. By design, the system must be able to limit the scope of an element’s actions, which has two desirable effects: the impact of a failure, corruption, or misuse of the element will be minimized; and the analysis of the element will be simplified. A design driven by least privilege considerations results in sufficiently fine granularity of privilege decomposition and the ability for fine-grained allocation of privilege to human and machine elements.

A strategy for application of least privilege can be summarized as allocate the minimal (separate) privileges for an entity according to the extent to which that entity has a "need-to-perform", where "perform" can be replaced by a verb (e.g., know, modify, delete, use, configure, authorize, start/enable, stop/disable, and so on) [25].

In addition to its manifestations at the system interface, least privilege can be used as a guide for the internal structure of the system itself, such as how to use domain separation. One aspect of internal least privilege is to construct modules so that only the elements encapsulated by the module are directly operated upon by the functions within the module. Elements external to a module that may be affected by the module’s operation are indirectly accessed through interaction with the module that contains those elements.

Ref: [17][25][41][49]

7.12 Least Sharing

System resources should be shared among system elements only when necessary, and among as few system elements as possible.

Elaboration:

Sharing via common mechanism and other means can increase the susceptibility of system resources (data, information, system variables, interfaces, functions, services) to unauthorized access, disclosure, use, or modification and can adversely affect the capability provided by the system. Minimized sharing also helps to simplify the design and implementation.

Any shared mechanism (especially one involving shared variables) represents a potential information path between system elements, and the design should account for ensuring such interaction does not unintentionally violate the design intent. A design that employs least sharing helps to 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.

Two criteria provide the basis for application of least sharing:

- Share only if absolutely necessary: This is a trade decision that factors in the cost-benefit of sharing against the increased exposure that results from the sharing.
- Minimize any sharing that is allowed: This is a constraint on the extent of sharing.

Least common mechanism, a historically well-known security design principle, is an instance of least sharing.

Also referred to as: Minimized Sharing

Ref: [17][41][49]

7.13 Loss Margins

The system is designed to operate in a state space sufficiently distanced below the threshold at which loss occurs.

Elaboration:

"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 the operational conditions and the associated adversity level at some “distance” (i.e., conservative threshold) away from the estimated critical adversity threshold or loss-triggering threshold. Loss margins also allow increased time to detect the need for response action (see Anomaly Detection), to determine what the response action should be (see Commensurate Response), and to complete the selected response action. When uncertainty about response action effectiveness may exist,

loss margins need to allow time to evaluate response effectiveness, determine any additional actions needed, and complete any selected actions.

Loss margins are effective in addressing uncertainty about how and when a loss triggering event occurs. Specifically, loss margins are effective to address uncertainty associated with

- intelligently designed and executed attacks, to include attacks that persist and evolve overtime, and
- unknown, unquantified, and underappreciated susceptibility, threat, hazard, vulnerability, and the associated risk.

Uncertainty may derive from the environment of operation, the design and realization of the system, the utilization and sustainment of the system, and the adversity presenting itself to the system.

For designs incorporating loss margins, uncertainty about adversity makes determining the loss-triggering thresholds difficult. Loss margins for design should be determined with a balance between certainty (what has happened and can happen again) and uncertainty (what has not happened but can happen, what has happened but can also happen in a different way than before). 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 the margins. Other factors for computing loss margins include system complexity, use of newer technology or older technology in new ways, and 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 all actual, simulated, or emulated adversity necessitate larger margins to account for the remaining uncertainty. The size of the margin may be reduced with time as unknown and underappreciated loss scenarios are uncovered and corrected, or the size may need increasing over time as malicious adversity capability matures in sophistication.

The term “loss margins” is synonymous with the term “loss reserves”.

Ref: [16][33][34][35][1][23]

7.14 Mediated Access

All access to and operations on system elements are mediated.

Elaboration:

Mediated access has two parts: (1) a policy-based access mediation decision and (2) the enforcement of the access mediation decision. The access decision may include conditional constraints that further restrict the access. These constraints include role, time of day, system state or mode, or duration of operation.

If access is not sufficiently mediated, then there is no possibility of limiting how system elements (human and machine) interact to ensure that only authorized behaviors and intended outcomes result.

Mediated access is a foundational principle in the design of secure systems. Mediated access is achieved by an access mediation control mechanism. Seminal computer security work defined the *reference validation mechanism* as the generalized form of any mechanism that is an implementation of the *reference monitor concept.* The reference monitor concept provides the design assurance basis to demonstrate the trustworthiness of a mediated access control mechanism. NEAT provides a refinement to extend the reference monitor concept (see Section 6.8).

Mediated access has the purpose to achieve the following:

- Place limits on access to, and use of, the system;
- Reduce the possibility of loss escalation; and
- Reduce the extent to which loss escalates and propagates.

Mediated access is based on the interaction between an entity and a target element, and has two aspects:

1. access to the element: requesting entity has only authorized access to a target element
2. use of the element. Requesting entity is allowed to perform only authorized operations on the target element

Mediated access may enforce distributed privilege, least privilege, and least sharing constraints.

“Efficiently mediated access" refers to using least common mechanism for mediating access. Mediating access is often the predominant security function within a secure system and if not designed correctly, may result in performance bottle necks. Use of least common mechanism is one means to help reduce bottle necks [49].

Ref: [17][41][49][39][20][6]

7.15 Minimize Detectability

The design of the system should minimize the detectability of the system as much as practicable.

Elaboration:

A system that is not exposed to, or discoverable, observable, or trackable by an adversarial threat, is less prone to a targeted attack. Minimizing detectability forces attention to eliminate or reduce exposures such as unnecessary interfaces, access points, footprints, and emanations, thereby reducing the susceptibility to adversarial threat actions.

Interfaces and access points have the effect of exposing the system to intentional (attack) and non-intentional (fault, error, incident, accident, etc.) adversity. Yet interfaces and access points are necessary to compose system elements to deliver required capability, and some duplication of interfaces and access points is needed to avoid single points of failure. System design must balance the need for interfaces with the susceptibility resulting from the interface being exposed, discovered, and observed. Every interface, internal or external, constitutes an exposure that must be accounted for.

Minimizing detectability reduces the ability of an adversary to observe and discover information about the system to craft and execute attacks. This includes detection of system location, presence, and movement due to emissions, signature, footprint, etc.

Some ways a system may be detectable include heat emission, electronic magnetic (EM) emissions, sound, or vibrations; reflects radar waves or light; or the response to stimulus (e.g., responds to an Internet Control Message Protocol (ICMP) echo request or “ping”). Camouflage, stealth, low probability of intercept/low probability of detect (LPI/LPD) waveforms (for radios), and frequency hopping are all specific forms or means to minimizing detectability.

This principle may be considered a generalization of “reducing the attack surface”.

Ref: [50][43][46]

7.16 Protective Defaults

The default configuration of the system provides maximum protection effectiveness.

Elaboration:

The configuration of the system encompasses 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 'out-of-the-box’ system configuration and parameters aggressively prioritize the achievement of loss control objectives over the ability to deliver 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.

See also in literature: secure defaults, Fail-safe defaults

Ref: [17][41][49]

7.17 Protective Failure

A failure of a system element should neither result in an unacceptable loss, nor invoke another loss scenario.

Elaboration:

Protective failure is the aspect of continuous protection ensuring protection capability is not interrupted during a failure and the effect of the failure is constrained.

The two aspects of protective failure must be satisfied to achieve the intended effect.

- Avoid single point of failure: The failure of a single system element should not lead to unacceptable loss; unacceptable loss should only occur only in the case of multiple independent malfunctions – a safety principle known as “single failure criterion.”
    - Defense-in-depth is an approach that helps 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.

- Defense-in-depth does not address the propagation of failure by invoking a new loss scenario and therefore does not help achieve this aspect of protective failure without additional analysis.

Protective failure applies to discrete elements, aggregates of elements, and to the systems abstraction. Protective failure seeks to limit the effect of a failure to the extent practicable, and in doing so minimize the introduction of new loss possibilities. Protective failure is able to limit the extent to which a failure is able to (a) advance loss scenarios associated with the failure, including cascading losses; (b) trigger a different loss scenario; or (c) create a new loss scenario.

Efforts to avoid or limit failures may themselves degrade system performance, a form of failure. Thus, system designers may need to consider trade spaces between possible adverse effects and system performance.

Concept rolls up: Secure Failures, Fail-safe/Safe Fail, Loose Coupling

Ref: [41][47][16][33][49]

7.18 Protective Recovery

The recovery of a system element should not result in, nor lead to, unacceptable loss.

Elaboration:

Protective recovery is an aspect of continuous protection that ensures that protection capability is not interrupted during the recovery from actual or impending failure. Protective recovery must be applied to discrete elements, aggregates of elements, and to the overall system. To the extent practicable, any recovery from impending or actual failure to resume normal, degraded, contingency or alternative operation, or recovery of other asset losses, should not (a) advance the loss scenario that is the target of the recovery, (b) trigger other loss scenarios, or (c) create new loss scenarios. The practicable aspect of this recognizes that for some recovery efforts to be successful, they may themselves 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 the expediency in the need for a protective recovery.

Ref: [25][41][34][49]

7.19 Redundancy

The design delivers the required capability by the replication of functions or elements.

Elaboration:

Redundancy employs multiples of the same elements, data and control flows or paths to avoid single points of failure. Redundancy requires a strategy for how the multiple elements are used individually or in combination (load-balancing; concurrently; fail-over; backup; voting, agreement, consensus).

Redundant solutions are susceptible to common mode failure - a single event that results in the same or equivalent elements failing in the same manner. The cause may occur with intent or without intent. Diversity is a means to address concerns of common-mode-failure.

Ref: [25][41][47][33]

8 Summary

This report provides a set of multidisciplinary principles and supporting terms and definitions for systems engineering design of trustworthy cyber-physical systems (CPS), with emphasis on controlling the adverse effects (i.e., loss) that *might occur* as a direct or indirect result of the system delivering specified capability at specified levels of performance.

The principles and terms are representative of the practices of the safety, security, survivability, and resilience communities and specialties, and encompass the concepts and theorems from the disciplines of computational science, control systems, systems engineering, software engineering, fault/failure tolerance, and mathematics.

The principles and terms are intended to be used as a starting point for discussion, vetting, employment, and ultimately acceptance by the engineering community. Their acceptance facilitates standardization within and across the general and specialty systems engineering practices and provides content suitable for inclusion in engineering bodies of knowledge.

The design principles are structured in the following three categories with the combined intent to prevent the occurrence of an adverse effect (loss) to the extent practicable, and for those cases where an adverse effect (loss) does occur, to limit the extent of the adverse effect (loss).

- Engineering design principles fundamental to managing complexity
- Trustworthiness design principles fundamental to the design of a system for which there is justified confidence in its ability to function and produce outcomes only as intended
- Loss control design principles fundamental to achieving loss control objectives

These principles can be applied immediately due to their basis in existing specialty engineering practices. However, the there is also the opportunity for the refinement and extension of the principles as part of the formalized digital representations of requirements, architecture, and design used in Digital Engineering (DE) environments and tools.

Appendix A References

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

[2] D. A. Simovici and C. Djeraba, "Partially Ordered Sets", in Mathematical Tools for Data Mining: Set Theory, Partial Orders, Combinatorics, Springer, 2008.

[3] Department of Defense, Defense Standardization Program, Standardization Directory, SD1, 1 April 2019.

[4] Department of Defense, Department of Defense Handbook: Systems Requirements Guidance, MIL-HDBK-520A, Washington DC, 2011.

[5] Department of Defense, Department of Defense Standard Practice: System Safety, MIL-STD-882E, Washington DC, 2011

[6] Department of Defense, Department of Defense Trusted Computer System Evaluation Criteria (TCSEC), DoD 5200.28-STD, December 1985

[7] Department of Defense, General Guidelines for Electronic Equipment, MIL-HDBK-454B, 15 April 2007

[8] Department of Defense, Technology and Program Protection to Maintain Technological Advantage, DODI 5200.83, Washington DC, 20 July 2020.

[9] G. M. Uchenick and W. M. Vanfleet, "Multiple independent levels of safety and security: high assurance architecture for MSLS/MLS," MILCOM 2005 - 2005 IEEE Military Communications Conference, Atlantic City, NJ, 2005, pp. 610-614 Vol. 1

[10] H. Sillitto, J. Martin, D. McKinney, R. Griego, D. Dori, D. Krob, P. Godfrey, E. Arnold, and S. Jackson, "Final SE definition", 8 January 2019. [Online]. Available: https://www.incose.org/docs/default-source/default-document-library/final_-se-definition.pdf?sfvrsn=340b9fc6_0. [Accessed 27 July 2020].

[11] H. W. Jones, "Common Cause Failures and Ultra Reliability", [Online]. Available: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20160005837.pdf. [Accessed 27 July 2020].

[12] INCOSE, Systems Engineering Body of Knowledge (SEBOK), Part 6: Related Disciplines/SE and Specialty Engineering/System Resilience, http://sebokwiki.org/wiki/System_Resilience. 2019

[13] ISO/IEC/IEEE, ISO/IEC/IEEE 15026:2014 Systems and Software Engineering - Systems and Software Assurance - Part 1: Concepts and Vocabulary, 2014.

[14] ISO/IEC/IEEE, ISO/IEC/IEEE 15288:2015 Systems and software engineering – System life cycle processes, 2015.

[15] ISO/IEC/IEEE, ISO/IEC/IEEE 24765-2017 Systems and software engineering – Vocabulary, 2017.

[16] J. H. Saleh, K. B. Marais, and F. M. Favaro, "System safety principles: A multidisciplinary engineering perspective", Journal of Loss Prevention in the Process Industries, vol. 29, pp. 283-294, 2014.

[17] J. H. Saltzer and M. D. Schroeder, "The Protection of Information in Computer Systems", Proceedings of the IEEE, vol. 63, no. 9, pp. 1278-1308, September 1975.

[18] J. H. Saltzer and M. F. Kaashoek, “Principles of Computer System Design”, 2009.

[19] J. L. Bayuk and B. M. Horowitz, “An Architectural Systems Engineering Methodology for Addressing Cyber Security”, Systems Engineering, pp. 294-304, 16 February 2011.

[20] J.P. Anderson, “Computer Security Technology Planning Study”, ESD-TR-73-51 Volume 1, Deputy for Command and Management Systems, HQ Electronic Systems Division (AFSC), October 1972

[21] Joint Chiefs of Staff, Joint Publication 3-0, Joint Operations, 22 October 2018.

[22] Joint Chiefs of Staff, Joint Publication 3-12, Cyberspace Operations, 8 June 2018.

[23] L. P. Pagani, “On the Quantification of Safety Margins”, PhD Dissertation, Massachusetts Institute of Technology, September 2004.

[24] M. Bishop, Computer Security – Art and Science, Addison-Wesley, ISBN 0-201-44099-7, 2003

[25] M. D. Schroeder, D. D. Clark, and J. H. Saltzer, “The Multics Kernel Design Project”, in Proceedings of Sixth ACM Symposium on Operating Systems Principles, 1977.

[26] M. D. Watson, “Systems Engineering Principles and Hypotheses”, INCOSE INSIGHT, vol. 22, no. 1, pp. 18-28, May 2019.

[27] M. McEvilley, G. Vecellio, “Strategic Vision for Safety and Security in Weapon Systems Engineering”, MITRE Technical Report/MTR 180261, July 2018

[28] M. Schroeder, “Cooperation of mutually suspicious subsystems in a computer utility”, Ph.D. dissertation, M.I.T., Cambridge, Mass., 1972

[29] M. T. Span, L. O. Mailloux, M. R. Grimaila and W. B. Young, “A Systems Security Approach for Requirements Analysis of Complex Cyber-Physical Systems”, 2018 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), Glasgow, 2018, pp. 1-8

[30] M. Winstead, “An Early Attempt at a Core, Common Set of Loss Driven Systems Engineering Principles”, INCOSE INSIGHT, vol. 23, no. 4, December 2020.

[31] Merriam-Webster, Dictionary by Merriam-Webster, 2020. [Online]. [Accessed 28 July 2020].

[32] N. G. Leveson, “Engineering a Safer World – Systems Thinking Applied to Safety”, Chapter 14, MIT Press, ISBN 978-0-262-01662-9, 2011

[33] N. Moller and S. O. Hansson, “Principles of Engineering Safety: Risk and Uncertainty Reduction”, Reliability Engineering & System Safety, vol. 93, no. 6, pp. 798-805, June 2008.

[34] National Aeronautics and Space Administration (NASA), System Safety Handbook Volume 1: System Safety Framework and Concepts for Implementation, NASA/SP-2010-580, Version 1.0, November 2011

[35] National Aeronautics and Space Administration (NASA), System Safety Handbook Volume 2: System Safety Concepts, Guidelines, and Implementation Examples, NASA/SP-2014-612, Version 1.0, November 2014

[36] National Aeronautics and Space Administration (NASA), Systems Engineering Handbook Rev 1, NASA/SP-2007-6105, Dec 2007.

[37] O. Sami Saydjari, Engineering Trustworthy Systems – Get Cybersecurity Design Right the First Time, McGraw-Hill, ISBN 978-1-260-11817-9, 2018

[38] Oxford English Dictionary, Oxford University Press, 2020. [Online]. Available: https://www.oed.com/. [Accessed 27 July 2020].

[39] P. G. Neumann, “Fundamental Trustworthiness Principles”, 2017.

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

[41] P. G. Neumann, “Principled Assuredly Trustworthy Composable Architectures”, Menlo Park, California: SRI International, 2004.

[42] R. Adcock, S. Jackson, J. Singer and D. Hybertson, “Principles of Systems Thinking”, Stevens Institute of Technology, 10 May 2020. [Online]. Available: https://www.sebokwiki.org/wiki/Principles_of_Systems_Thinking. [Accessed 27 July 2020].

[43] R.E. Ball, “The Fundamentals of Aircraft Combat Survivability Analysis and Design”, 2nd Edition. AIAA Education Series. pp. 2, 445, 603. ISBN 1-56347-582-0

[44] R. E. Smith, “A Contemporary Look at Saltzer and Schroeder's 1975 Design Principles”, IEEE Security & Privacy, vol. 10, no. 6, pp. 20-25, November/December 2012.

[45] R. Ross, M. McEvilley, J. Oren, NIST SP 800-160 Volume 1: Systems Security Engineering: Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems, November 2016, includes updates as of 03-21-2018

[46] R. Ross, V. Pillitteri, R. Graubart, D. Bodeau and R. McQuaid, NIST SP 800-160 Volume 2: Developing Cyber Resilient Systems: A Systems Security Engineering Approach, November 2019

[47] S. Jackson and T. Ferris, “Resilience Principles for Engineered Systems”, Systems Engineering, vol. 16, no. 2, pp. 152-164, 15 July 2013.

[48] S. Sheard, M. Konrad, C. Weinstock, and W. Nichols, “A Complexity Measure for System Safety Assurance”, in INCOSE International Symposium, Adelaide Australia, 2018.

[49] T. E. Levin, C. E. Irvine, T. V. Benzel, G. Bhaskara, P. C. Clark and T. D. Nguyen, “Design Principles and Guidelines for Security”, Naval Postgraduate School, 2007.

[50] W.D. Bryant and R.E. Ball, “Developing the Fundamentals of Aircraft Cyber Combat Survivability: Part 2”, Joint Aircraft Survivability Program Office, Aircraft Survivability Journal, Spring 2020

Appendix B Glossary

The principles in this report were compiled from sources representing various security and non-security disciplines. As such, it was necessary to provide an extensive glossary to aid in the interpretation of the principles, while also offering elaboration as needed to provide additional perspectives and context-sensitive interpretation.

Additionally, many terms used in the literature on principles for security and related disciplines have, frustratingly, been used without clear definition, or over time 'evolved' and not remained clear. For these terms, we have derived their meanings from context across [17][25][41][49][44][39].

Finally, to support the standardization of cyber-physical system engineering practice in response to the security, resilience, and survivability concerns associated with contested cyberspace, each term includes reference(s) when used unmodified or with some adaptation, or the key reference(s) inspiring the definition.

Abstraction	view of an object that focuses on the information relevant to a particular purpose and ignores the remainder of the information [15]
Access	the ability to make use of a resource (e.g., data or function); to obtain the use of a resource [17][39]
	Note: access is typically not all or nothing, e.g., access to data may be read only; may be read or modify existing content; or may be the ability to read, modify, create, or delete content.
Adversary	a party acknowledged as potentially hostile to a friendly party and against which the use of force may be envisaged. [21]
	Note: the broadest interpretation of party is an individual, group, organization, or nation-state entity
Adversity	anything that can contribute to or result in a loss
	Note: adversity can be intentional (malicious) or unintentional (non-malicious)
	Note: adversity is characterized by terms that include attack, hazard, susceptibility, threat, vulnerability
Anomaly	something that deviates from what is considered or accepted as standard, normal, or expected [15]
	Note: anomalies include erroneous, unexpected, unpredicted, unknown behavior or outcomes. An anomaly may result from intentional or unintentional adversity.
Asset	an item of value to stakeholders. [15][45]
	Note: Assets are broadly categorized as tangible (e.g., a physical item such as hardware, firmware, computing platform, network device, other technology component, or humans) or intangible (e.g., data, software,

capability, function, service, trademark, copyright, patent, intellectual property, image, or reputation).

- Assets may be defined or managed as individual items or as an aggregate or groups of items (e.g., personnel data; fire control function; environmental sensor capability).
- Assets may have a lifetime that is distinct from but may coincide with aspects of the system life cycle.
- Assets include military equipment (aircraft, ships, installations, either employed or targeted) [15]
- Assets may 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).

Asset definition and valuation is determined by stakeholders and may vary across stakeholders. Asset value is relative to an expressed need, expectation, or association. These considerations include

- Effect of degradation or destruction
- Cost to repair or replace
- Cost to achieve objectives supported by the asset using other means
- Mission context
- Operational context
- System context (state/mode)

Assurance grounds for justified confidence that a claim has been or will be achieved [13]

Assured to achieve the desired level of assurance [39][13]

Attribution the ability to trace an effect on the system state, behavior, outcomes, or environment to an initiating entity [11]

Notes: The trace is evidence-based and may include various forms of information that is an aggregate of direct and indirect data and meta-data.

To avoid wrongful attribution in loss scenarios with extremely high consequences necessitates having high confidence in the underlying evidence and its tractability.

Authorize to grant use of, or access to, a resource; to explicitly allow a particular behavior [17][39]

Cohesion the manner and degree to which the tasks performed by a single module are related [39]

Common cause failure: a failure occurring when one event or shared factor causes two or more failures [11]

Note: Because common cause failures have the same cause, the failures are not statistically independent. This is contrary to the core assumption of usual reliability theory, that all failures are independent and uncorrelated.

Common mode failure: a common cause failure where several elements fail in the same way due to the same reason [11]

Note: Common mode failure may result from intentional action (an attack) that exploits the common vulnerability of redundant subsystems, or from unintentional action (environment, human, or system) that triggers a vulnerability common to redundant components.

Common mode failure reduces the effectiveness of redundancy.

Compromise (verb) to cause an unauthorized event; (noun) the unauthorized result of an event

Note: Compromise can occur with intent (typical use whereby some action is taken with express purpose) or without intent (an unintended error can have the same result as an intentional action).

Compromise may result from an adversary establishing a presence in the system and then using the system in a manner that is inconsistent with the design intent for the system.

In this case compromise reflects undesired behavior,

Note: Compromise has the following general forms:

- an element of the system is counterfeited, modified, substituted, replaced, or altered with the result that the element does not meet its performance specifications
- an element, subsystem, or the system itself is used, leveraged, or controlled (to include overloaded, over utilized) in a manner that results in behavior (direct, emergent, side-effects) other than the design intent.

Examples of typical uses of the term compromise include [37]:

- Compromise (data): the unauthorized disclosure of sensitive data
- Compromise (system): the unauthorized modification of data or of the system

Coupling manner and degree of interdependence between modules [39]

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

Notes: 'Separating domains' becomes a means to compose the system in a manner that helps enforce separation of roles, separation of privilege (enforce least privilege), the rules governing their use, or satisfy protection needs.

Various stakeholders often having varying concepts and definitions for ‘domain,’ for example: classification domains, technological domains,

	authoritative domains, functional domains, etc. Domains are a highly multidimensional construct where pertinent domains depend on stakeholder priorities and loss control objectives.
Element	a discrete part of a system that can be implemented to fulfil specified requirements [14]; an identifiable part, one of the parts of a compound or complex whole [39] Note: A system element can be hardware, software, data, humans, processes (e.g., processes for providing service to users), procedures (e.g., operator instructions), facilities, materials, and naturally occurring entities (e.g., water, organisms, minerals), or any combination [14]
Entity	a subject (active entity) or object (passive entity) with a distinct and independent existence that acts, accesses, is acted upon or is accessed. Note: • An entity can be an element, process, or data (though data may only be an object). • An entity can be 'acted through': A acts directly on B, and B acts directly on C, thus A acts on C through B Interactions may be of the form subject-to-subject (e.g., one process interacting with another process) and subject-to-object (e.g., one process acting upon an object).
Environment	context determining the setting and circumstances of all influences upon a system [39] Note: including the physical domains of air, land, maritime, space and the information domain of cyberspace. [22]
Failure	termination of the ability of an element or system to perform a required function or its inability to perform within previously specified limits and specifications [13] Note: erroneous system function (i.e., no longer behaves per its design intent), disruption of system function (i.e., intermittent correct function, gaps in correct function) or undesired termination of system function (i.e., ceases to function at all)
Hazard	a real or potential condition that could lead to an unplanned event or series of events (i.e., mishap) resulting in death, injury, occupational illness, damage to or loss of equipment or property, or damage to the environment. [5] Note: Preventing or otherwise limiting hazardous conditions during system design is the goal of the loss control objectives, i.e., to control loss. Note: A hazard is a description of conditions that may lead to loss. The removal of the conditions or exposure of a system to the conditions eliminates loss potential; understanding the conditions informs engineering to limit loss when the removal of conditions or exposure to the conditions is not feasible or practical. This is synergy in the goal to

	control loss and the focus of the system safety design order of precedence to control hazards and associated mishaps, thereby also controlling loss. Note: A security interpretation of hazard adds damage to or loss of data, information, or capability, and shifts the focus from the mishap to a loss.
Isolation	limiting or cutting off shared elements and resource flows among elements, components, subsystems, or systems. Complete isolation would have no sharing or resource flows.
Layer	group of functionally or conceptually related modules partitioned from other layers/groups of modules [25][41]; a partition resulting from the functional division of a system, where layers are organized in a hierarchy and there is only one layer at each level in the hierarchy [39]
Least:	to the smallest extent possible Note: "Least" refers to what has a necessity to achieve to the smallest extent possible. In some cases, it would be reflective of an 'objective' requirement.
Loss	The state of no longer having an asset, no longer having as much of an asset, having an undesired change in the condition of an asset, or having an undesired behavior of an asset. Note: The specific interpretations of loss are fully dependent on the type of asset. For example: the interpretation of loss for a person differs from the loss interpretation for a system function, which differs from the loss interpretation for data and information.
Margin	a spare amount or measure or degree allowed or given for contingencies or special situations [31]. The allowances carried to account for uncertainties and risks. Note: There are two general forms of margins: • Design margin: Margin allocated during design based on assessments of uncertainty and unknowns. This margin is often consumed as the design matures. [36] • Operational margin: The operational margin that is designed-in explicitly to provide “space” between the worst normal operating condition and the point at which failure occurs (derives from physical design margin) [10][34]
Mediate	to check for authority, especially authority to access; to validate access privileges and grant access [17]
Minimal	smallest degree or amount practical Note: Often synonym of least but 'minimal' often used to refer to less 'absolute' terms. Where applicable, 'minimal' refers to a threshold condition, with 'least' the objective condition
Minimize	reduce to the smallest possible or practical amount or degree

Module	a well-defined 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. [25][39][15]
Object	a passive entity that contains or receives information. Access to an object potentially implies access to the information it contains. Examples of objects: records, blocks, pages, segments, files, directories, directory trees, and programs, as well as bits, bytes, words, fields, processors, video displays, keyboards, clocks, printers, network nodes, etc. [6]
Partially ordered set	a set together with a binary relationship where any two members are either unrelated or one precedes the other in the relationship, but no two members precede each other, i.e., if a < b then b < a is not true (antisymmetric), where "<" represents the preceding relationship. Additionally, if a < b and b < c, then a < c (transitivity) [2]
Recover	to re-establish an asset in a state or form whereby it is able to fulfill its intended purpose Note: Recover is the response action to a loss condition of an asset. The interpretation of "to re-establish" is therefore based on the type of asset and the specific case of loss of that asset. While the ability to recover is typically considered in terms of a capability, function, or service, it also applies to re-establish data and information in a usable form. Note: In all cases, the need to re-establish an asset may result from some partial loss or total loss (e.g., destruction) of the asset.
Reference Monitor Concept	an abstract model of the necessary and sufficient properties that must be achieved by an access mediation control mechanism. Note: Mediated access is the primary underpinning of any protection capability of a system. The reference monitor concept states the design assurance basis for the trustworthiness of mediated access control functions. An access mediation control mechanism has two parts: (1) a policy-based access mediation decision, and (2) the enforcement of the access mediation decision. The reference monitor concept does not refer to any particular access mediation policy, nor does it address any particular implementation of the mechanism. The reference monitor concept provides assurance in approaches to the design and development of security mechanisms that enforce an access mediation policy. The reference monitor concept reflects an "ideal mechanism" characterized by three properties: • the mechanism is tamper-proof (i.e., it is protected from modification so that it is always capable of enforcing the intended access control policy); • the mechanism is always invoked (i.e., it cannot be bypassed so that every access to the resources it protects is mediated); and

- the mechanism can be subjected to analysis and testing to assure that it is correct (i.e., it is possible to validate that the mechanism faithfully enforces the intended security policy and that it is correctly implemented).

Note: A generalization of the reference monitor concept serves well as the design basis for individual system elements, collections of elements, networks, and systems where intentional and unintentional adversity drives the need for control functions to achieve loss control objectives. The concept also drives the need for rigor in engineering activities that is commensurate to the trustworthiness sought in system elements.

Ref: [20][41][49]

Reference Validation Mechanism — that combination of hardware and software which implements the reference monitor concept. [20]

Resilience — the ability to provide required capability in the face of adversity. [12]

Resources — any usable part of a system that is controlled and assigned, or items/assets used or consumed during execution of a function. [14]

Rigor — quality of being detailed, careful, complete, and thorough

Note: Rigor is a means to achieve assurance. Rigor identifies the formality, thoroughness, accuracy, and precision to be applied in systems engineering approaches, methods, and outcomes, all of which lead to a corresponding level of confidence.

Safety — freedom from conditions that can cause death, injury, occupational illness, damage to or loss of equipment or property, or damage to the environment. [5]

Security — freedom from those conditions that can cause death, injury, or occupational illness; damage to or loss of equipment or property, damage to the environment; damage or loss of data or information; or damage to or loss of capability, function, or process. [adapted [36][5][45]]

State (Mode) — condition of existence that a system, component, or simulation can be in [39]

Notes: State and mode are often used interchangeably whereby there can be states with modes or modes with states. The distinction between states and modes is arbitrary. A system or subsystem may be described in terms of states only, modes only, states within modes, modes within states, or any other scheme that is useful. [4]

States and modes can be determined at the system level, element level, and function level. Examples of system states and modes include idle, ready, active, training, degraded, emergency, backup, wartime, peacetime, battle short. Functional states and modes may be a phase or segment of operation, for example takeoff, climb, cruise, descent, approach, landing; or a phase or segment of a sequence, for example, enable, arm, fire.

Defining specific states and modes is often motivated as a means of partitioning (enabling) specific system functions and capabilities within the enumerated states or modes.

State Space the set of all possible states of existence a system may be in [15][42]

Note: The State Space is a useful abstraction for reasoning about the behavior of a given system. For example, the analysis of 'loss margins' may be enabled via analysis of State Space subsets in terms of desired and undesired conditions.

Subject an active entity, generally in the form of a person, process, or device that causes information to flow among objects or changes the system state. [6]

Survivability the ability to exist or to function despite adversities in an environment [38][50][43]

Note 1: Survivability requires context for interpretation. Survivability contexts include the mission, the system, human occupants of the system or humans that interact with the system.

Note 2: Temporal constraints may bound the definition and assessment of what is means to be survivable.

Note 3: The environments for survivability include the physical environments of air, land, maritime, and space, and the virtual environment of cyberspace.

Note 4: Survivability encompasses the achievement of three goals: avoid being hit; withstand the effect of being hit; and aptly recover from any unacceptable loss effects when hit.

The term "hit" may refer to being struck or impacted by a kinetic weapon (e.g., missile, torpedo, electromagnetic pulse) or cyber weapon. A cyber weapon is said to have 'hit' a target if the weapon accesses and somehow modifies the system, including activating one or more implanted malfunction mechanisms ("malware") [50].

Note 5: "function" is interpreted in terms of mission, ability to support required safety means, and the capability of the system itself. For example, for combat aircraft, survivability may be in terms of completing mission and returning to base safely [50]. If an attack on an aircraft disables or degrades systems that enable safe landing but otherwise does not impact the aircraft, mission may be completed but the aircraft crashes upon return, failing to achieve survivability. Full survivability includes completing mission and preserving the systems and operator life; lesser survivability still requires safety for operators.

Note 6: In weapon system engineering, survivability encompasses the following attributes, which provide a survivability correlation to, and interpretation of, the loss control objectives [43]:

- Susceptibility – the inability to avoid being hit (by adversity).
  - Prevent a loss from occurring

- Vulnerability – the inability to withstand the hit.
  - Limit the extent of loss
- Recoverability – ability to partially or fully *recover* from the loss effects of being hit
  - Recover from a loss

Note 7: For a system to be survivable despite diverse adversities ultimately depends on a transdisciplinary design approach involving numerous specialties (e.g., reliability, safety, security, resilience) with consideration of incidents, accidents, malice, and margins for processing, bandwidth, data storage, and other attributes. [41][40].

Susceptibility the inability to avoid an adversity. [43][50]

Note: the adversity may be natural or man-made and hostile or non-hostile

Note: The problem is the inability to avoid adversity. Engineered systems may enter the environment and still avoid specified adversities while in the environment. So, susceptibility is when the engineered system is unable to avoid an adversity while in the environment. The greater the potential that a system may "be hit" the more susceptible it is.

System an arrangement of parts or elements that together exhibit behavior or meaning that the individual constituents do not [10]

Notes: An engineered system is a system designed or adapted to interact with an anticipated operational environment to achieve one or more intended purposes while complying with applicable constraints.

Alternatives [39]: 1. combination of interacting elements organized to achieve one or more stated purposes; 2. product of an acquisition process that is delivered to the user; 3. something of interest as a whole or as comprised of parts; 4. interacting combination of elements to accomplish a defined objective (or [14] – combination of interacting elements organized to achieve one or more stated purposes).

Threshold the entry/starting point or level where an acceptable or unacceptable condition is initially reached [26][39]

Traceability the ability to determine all the entities and elements involved in specific system behaviors, interactions, and outcomes [39][41]

Note: Traceability requires a dependable infrastructure that can record all relevant details associated with the system delivering required capability. Traceability is dependent on a unique identity of all entities and elements to conclude with confidence which entity performed an action independently or on behalf of some other entity. Traceability requires that all relevant sequence of actions that are conducted are recorded and are protected from unauthorized access and modification.

Trust (noun) belief that an entity meets expectations; (verb) to believe that an entity meets expectations [[38], adapted]

	Note: Trust is a belief, and fundamentally, trust merely implies that you trust an entity whether the entity is trustworthy or not. A trustworthy entity is one for which sufficient evidence exists to make a conclusion of its trustworthiness. However, trust in an entity can occur without a basis or knowledge of trustworthiness. Trust may occur because 1) you have no alternative (e.g., trust the components involved in a transaction over the Internet without knowing the components that are involved in the transaction), 2) you do not even realize the necessity of trustworthiness, or 3) other reasons. [41] Trust is therefore not necessarily based on a judgement of trustworthiness, however the decision to trust an entity should consider the consequence, effect, and impact of the trust expectations not being achieved whether due to failure, deficiency, or incompetence. Ideally, criteria for trust are derived from the effect of non-performance.
Trusted Communication Channel	a mechanism that authenticates the endpoints involved in a communication and protects against data exfiltration (confidentiality, privacy) and data infiltration (integrity) while preserving the accuracy and precision (integrity) of the data in transit between the two endpoints. [41][49] Note: A trusted communications channel is typically a composition of trusted elements. Requirements usually include mediated access to the communication channel (to authenticate the endpoints involved in the communication and to ensure an acceptable match in their trustworthiness) and employing end-to-end protections for the data transmitted over the communication channel (to help protect against losses within scenarios involving interception and data modification, and to further increase the overall assurance of proper end-to-end communication).
Trusted Control Element	a trusted element with the specific purpose to 1) enforce a constraint on the system 2) make a change to the system configuration, function, or behavior and/or 3) provide a critical capability to achieving a control function
Trusted Element	an element shown to meet well-defined requirements under an evaluation of credible evidence Notes: Trust judgements should consider an evidentiary basis through all relevant system states, modes and transitions, accounting state/mode dependency and non-persistence within a state/mode. An element may be ‘trusted’ to meet one requirement but not necessarily all requirements allocated to it.
Trusted Path	a trusted communication channel between a human entity and a trusted authentication function of the system [25][6] Note: A trusted path provides assurance that 1) the human entity is communicating with the intended trusted authentication mechanism, and 2) the human entity has confidence, and therefore is able to trust that all

data and information exchanged (e.g., authentication credentials such as password or PIN) is received in unmodified form. Like a trusted communications channel, a trusted path ensures that all data and information exchanged is only known by the originator and intended recipient.

An example of a trusted path is many implementations of cntrl+alt+del to ensure the interaction between the user and the resulting login screen is trustworthy, as well as the resulting interaction between the screen inputs and the authentication function is trustworthy. Specifically, no third-party can intervene to capture the login credentials to use later to enter for illegitimate access.

Trustworthiness well-founded assessment of the extent to which a given system, network, or component will satisfy its specified requirements, and particularly those requirements that are critical to an enterprise, mission, system, network, or other entity. [41][49]

Note: 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. [41]

Consequently, trustworthiness is meaningful only with respect to those expectations. Reuse for a purpose other than the original may not be trustworthy for the new expectations.

Trustworthy worthy of being trusted to satisfy the given expectations [25][41][49]

Notes: an entity is trustworthy if there is sufficient credible evidence leading one to believe that the system will meet a set of given requirements [24].

A highly trustworthy element may be selected over a less trustworthy one in the case of cost-benefit factors other than trustworthiness. For such cases, any dependency of the highly trustworthy element upon a less trustworthy element does not degrade the overall trustworthiness of the resulting composition.

The conservative assumption is that the overall trustworthiness of a composition is that of its least trustworthy component or element. Trustworthiness of a particular composition may be greater than the conservative assumption; however, any such rationale should reflect logical reasoning based on a clear statement of the trustworthiness goals, as well as relevant and credible evidence. Such rationale would not include increased application of defense-in-depth layering within the composition, or replication of elements. Compositions to form trusted compound elements should have its dependencies form a partial ordering (i.e., if A depends on B, then B does not depend on A). Partial ordering provides a basis for trustworthiness reasoning and is essential regarding trustworthiness.

Note: An element (including a system) may be trustworthy to meet some but not all critical requirements allocated to it. For self-reliant trustworthiness, a system may be self-reliant trustworthy to meet some requirements but not self-reliant to be trustworthy for other requirements.

Vulnerability the inability to withstand adversity [43][50]

Note: the adversity may be natural or man-made and hostile or non-hostile

Note: This generalized perspective of vulnerability is based on the "withstanding being hit" concept of aircraft combat survivability and can be extended to the survivability concerns for all weapon system types since they are all susceptible to being attacked in/through the physical and cyberspace environments. This generalization is equally useful for security and resilience.

The generalization does not preclude equivalence with narrower and context-dependent definitions of vulnerability, such as the commonly used definition of vulnerability that is scoped to intentional threats and the protection of data and information systems (i.e., "weakness in an information system, system security procedures, internal controls, or implementation that could be exploited by a threat source", Committee on National Security Systems No. 4009).

Note: Vulnerability commonly occurs due to weakness(es) that can be intentionally exploited by an attack to achieve specific adverse effects or unintentionally triggered to cause adverse effects.

Note: The more likely it is that a system can suffer unacceptable loss when "hit", the more vulnerable it is.[50]