

Trustworthy Secure Operational Technology Systems

Strengthening NIST SP 800-82 Revision 4 Through Systems Security Engineering

Dr. Ron Ross

CEO, RONROSSECURE, LLC

Abstract

The draft fourth revision of NIST Special Publication 800-82 provides an essential and comprehensive foundation for managing cybersecurity risk in operational technology (OT). Its guidance on governance, asset management, access control, platform and infrastructure security, continuous monitoring, incident response, recovery, and OT architecture is necessary for every organization that depends on cyber-physical operations. Yet security controls, practices, and reference architectures alone cannot establish that an OT system will remain trustworthy secure when components fail, credentials are stolen, maintenance pathways are abused, complex supply chains are compromised, or adversaries obtain a foothold. Trustworthy secure systems concepts from NIST SP 800-160, Vol. 1, Rev. 1 add the missing engineering dimension: define unacceptable loss; reason about intended and emergent behavior; apply the Security Design Order of Precedence; use security design principles to constrain architecture; engineer for compromise; and produce evidence that justifies confidence in system-level claims. This paper describes practical, section-specific enhancements that can strengthen SP 800-82 without changing its purpose. The result is a constructive extension from protecting OT assets to engineering OT systems whose architecture bounds loss, preserves essential services, and supports safe, secure, and resilient operation throughout the system life cycle.

1. Introduction

Operational technology controls physical processes. Its failures can interrupt essential services, damage equipment, degrade product quality, harm the environment, and endanger human life. For that reason, the initial public draft of NIST SP 800-82, Revision 4 correctly treats safety, reliability, operational performance, and availability as integral to cybersecurity risk decisions. It gives owners, operators, integrators, vendors, and assessors a substantial body of practical guidance for identifying risk and implementing protections across diverse OT environments [1].

The publication is necessary. It is not, by itself, sufficient to build a *trustworthy secure* OT system. This is not a criticism of its scope or quality. SP 800-82 is principally guidance for managing OT cybersecurity risk and applying cybersecurity outcomes, controls, practices, and architectures. Building a system that can be trusted in adversity additionally requires the application of systems security engineering concepts, principles, and methodologies: a disciplined way to translate mission and stakeholder concerns into system requirements, architectural constraints, design decisions, verification evidence, and life-cycle assurance [2].

The distinction is consequential. A firewall can be configured correctly and still sit in an architecture with an unsafe bypass path. Multifactor authentication can protect remote access while a shared engineering workstation retains excessive authority. Redundant controllers can improve availability while sharing a common-mode software or power dependency. Monitoring can detect anomalies but arrive too late to prevent a physical process from crossing a hazardous threshold. Each control may work as specified, yet the composed system can still permit unacceptable loss.

SP 800-82 provides the essential cybersecurity foundation; trustworthy secure design turns that foundation into a system-level engineering argument.

Trustworthy secure systems concepts strengthen SP 800-82 by asking earlier and more structural questions: What losses are unacceptable? What system behaviors and states could produce them? Which capabilities or dependencies can be eliminated? Where must authority, information flow, and failure propagation be bounded? What should happen when protection fails? What objective evidence supports confidence in the resulting claims? Those questions complement, not replace, the safeguards and practices in SP 800-82.

Except where this paper directly describes or cites NIST guidance, the application of SP 800-160, Volume 1 concepts and principles to OT in the sections that follow represents the author's reasoned extension of that guidance; it should not be read as additional requirements stated by NIST in SP 800-160 or SP 800-82.

2. What SP 800-82, Revision 4 Already Establishes

Draft SP 800-82, Revision 4 is a strong platform for this extension. It places OT cybersecurity within enterprise risk management and governance; emphasizes operational consequences and requirements; addresses supply-chain risk; organizes outcomes around the core NIST Cybersecurity Framework functions of *Identify*, *Protect*, *Detect*, *Respond*, and *Recover*; expands OT architecture guidance; and provides an OT security control overlay and Risk Management Framework considerations [1]. Several features align especially well with trustworthy secure systems thinking.

- **Consequence awareness.** Section 3 recognizes that risk decisions must reflect physical, safety, environmental, financial, and mission consequences, not only data loss.
- **Architecture.** Section 5 discusses segmentation, management and operational networks, boundary devices, fault tolerance, redundancy, analog backups, and out-of-band management.
- **Life-cycle context.** The publication addresses legacy technology, maintenance, change, supply chains, incident response, recovery, and the constraints that distinguish OT from conventional IT.
- **Outcome orientation.** Section 4 describes desired cybersecurity outcomes while allowing organizations to tailor implementations to technology, risk, and mission context.
- **Control integration.** Appendices E and F connect OT considerations to NIST SP 800-53 and the Risk Management Framework, supporting governance, authorization, and assessment [6].

These are the right ingredients. The opportunity is to make their *engineering* relationships explicit. Revision 4 can more clearly show how consequence analysis generates protection needs; how those needs constrain architecture; how design principles inform structural choices; how controls realize the design; and how evidence supports system-level assurance.

3. The Remaining Engineering Gap

Cybersecurity guidance commonly describes what outcomes are desired and what safeguards may help achieve them. Engineering must also establish why a particular system, in its actual operating context, can be trusted to deliver those outcomes. The gap appears in four related areas.

First, risk descriptions do not automatically become engineered *requirements*. A statement that loss of view could harm operations must be translated into measurable constraints on data freshness, integrity, latency, independence, and failover behavior. Second, controls do not automatically compose into a trustworthy secure system. Interactions among controllers, safety systems, identity services, remote-access tools, field devices, and human operators can create behavior that is absent from any component viewed alone. Third, prevention cannot be assumed to succeed indefinitely. OT systems often remain in service for decades, incorporate components with limited security capability, and depend on multiple vendors and maintenance channels outside the direct control of the asset owner. The system architecture must therefore bound the *consequences of compromise*. Fourth, compliance evidence is not always assurance evidence. Showing that

a control exists is different from demonstrating that a critical enforcement path is non-bypassable, that a safe state remains reachable, or that recovery restores an authorized and known-good configuration.

A collection of protected components is not necessarily a trustworthy secure OT system; trustworthiness emerges from the components, their interactions, and the evidence behind system-level claims.

NIST SP 800-160, Volume 1, addresses this gap by treating security as an emergent property of the system and integrating security into systems engineering processes [2]. Volume 2 extends the approach to cyber resilience, including anticipating, withstanding, recovering from, and adapting to adversity [3]. *Engineering for Compromise* applies the same premise operationally: assume an adversary may penetrate some system component, then design the system so that compromise does not automatically produce system failure (i.e., mission loss) [4].

4. Begin with Mission, Behavior, State, and Loss

Trustworthy secure design starts with what the OT system must accomplish and what must never happen. Assets matter because of the loss their compromise can cause. For OT, unacceptable losses may include injury, release of hazardous material, loss of essential service, operation beyond certified limits, damage to long-lead equipment, corruption of product, or loss of the ability to observe and control a process [5].

The analysis should identify critical services and functions; intended system behaviors; unacceptable losses; hazardous or otherwise unacceptable system states; initiating conditions; control actions and information flows; and the time available to detect, contain, or recover. This shifts the emphasis from a generic threat list to the system's own loss potential. Threat intelligence remains important, but it informs rather than defines the protection problem.

This approach strengthens Section 3.1.3.1 and Section 4.1 of SP 800-82. Operational consequences become inputs to *requirements* and *architecture*, not merely factors used to rank risks. A high-consequence loss scenario should result in explicit claims such as: only the safety controller can issue a trip command; no remote-access session can change a safety set point; loss of the enterprise identity service cannot prevent local safe shutdown; or a compromised historian cannot alter control logic.

Table 1. From OT Concern to Engineering Claim and Evidence

OT concern	System-level claim	Representative evidence
Unsafe command	Only authorized control paths can issue high-consequence commands.	Architecture model; interface inventory; mediation and bypass tests.
Loss of view	Operators receive sufficiently fresh, authentic state data within the process decision time.	Timing analysis; independent sensor path; fault-injection results.
Remote compromise	A compromised support session cannot directly reach safety-critical functions.	Trust-boundary analysis; privilege map; segmentation and jump-host tests.
Common-mode failure	Essential control survives the loss or compromise of a shared service.	Dependency model; diversity analysis; failover demonstration.
Corrupt recovery	Recovery restores an authorized, known-good configuration before reconnecting to the process.	Signed baselines; restoration rehearsal; provenance records.

5. Treat Security as an Emergent System Property

System behavior emerges from composition. A controller, network switch, engineering workstation, authentication service, and remote-support platform can each meet its local requirements while their interactions create an unintended path to a critical function. This is particularly important in OT because digital interactions drive physical behavior, timing matters, and operators may intentionally bypass normal mechanisms during abnormal conditions.

SP 800-82, Revision 4 can strengthen its architecture discussion by requiring explicit reasoning across levels of abstraction. At each decomposition and integration step, systems engineering teams should identify trust boundaries, shared dependencies, information flows, assumptions, potential emergent behavior, authority paths, and failure propagation. The NIST SP 800-160, Volume 1 security design principles of *Structured Decomposition and Composition*, *Compositional Trustworthiness*, *Clear Abstractions*, *Reduced Complexity*, *Domain Separation*, and *Least Sharing* provide a vocabulary for this reasoning [2].

The explicit emphasis in SP 800-82, Revision 4 on applying zero trust principles creates another natural bridge to the security design principles in SP 800-160, Volume 1 [1]. Zero trust is not a substitute for systems security engineering or a synonym for network segmentation. It is strengthened when the principle of *Mediated Access* ensures that consequential interactions are authorized, the principle of *Least Privilege* limits the authority granted, and the principle of *Domain Separation* constrains the reach and effects of a compromised system element. Making this relationship explicit in Section 5 would connect the draft's zero-trust framing to system-level protection claims and evidence.

The practical product is not a more elaborate network drawing. Rather, it is a model that connects mission functions to system elements and shows which elements must be trusted, what they are trusted to do, what can interfere with them, and what happens when an assumption fails. This model should include cyber, physical, human, and enabling-system dependencies, including power, time sources, maintenance tooling, communications, cooling, software distribution, and recovery infrastructure.

6. Apply the Security Design Order of Precedence

SP 800-160's *Security Design Order of Precedence (SecDOP)* prioritizes loss-control alternatives. It first seeks to eliminate loss potential through design selection; then reduce the likelihood or severity of loss through design alteration; then incorporate engineered features or devices; then provide visibility and feedback; and finally rely on procedures, training, and protective equipment [2]. The order does not prohibit later measures. It prevents organizations from substituting operational burden for avoidable architectural weakness.

Applied to OT, SecDOP produces practical decisions. Eliminate an unnecessary remote-maintenance connection before adding another monitoring rule. Remove unused services and protocols before hardening them. Separate safety and basic process control domains before relying on access lists alone. Limit an engineering workstation's reachable assets and available commands before assuming user training will prevent error. Provide independent process indication before expecting a compromised human-machine interface to reveal its own deception.

The strongest OT safeguard may be a capability the system does not possess, a connection it does not expose, or an authority no single component holds.

SP 800-82, Revision 4 already recommends many compatible practices. SecDOP would give those practices a coherent priority and explain why architecture should precede compensating controls. It also offers a

repeatable decision record: for each important loss scenario, document which higher-order alternatives were considered, why they were feasible or infeasible, and what residual loss potential is addressed by subsequent measures.

7. Engineer for Compromise

Engineering for Compromise assumes that determined adversaries may evade prevention measures, exploit latent defects, misuse legitimate pathways, or compromise trusted suppliers. The design objective is not to concede defeat; it is to prevent a local compromise from becoming an unacceptable *operational loss* [4]. For OT, this means preserving essential control and safe operation even when selected system elements are untrustworthy.

Several security design principles are especially actionable:

- *Domain Separation* and *Least Sharing* limit propagation among enterprise, operations management, control, safety, vendor, and recovery domains. Separation should address identities, administration, data, software distribution, and common services, not only network zones.
- *Least Privilege* and *Distributed Privilege* restrict what a compromised identity or component can do. High-consequence changes may require independent authorization, physical presence, or concurrence from separately protected mechanisms.
- *Mediated Access* and *Continuous Protection* require every relevant interaction to pass through an enforcement mechanism whenever protection is needed. Maintenance and emergency pathways must not silently bypass the policy.
- *Minimal Trusted Elements* and *Reduced Complexity* shrink the set of components whose failure can defeat essential protection, making analysis and assurance more practical.
- *Protective Failure*, *Protective Recovery*, *Redundancy*, *Diversity*, and *Loss Margins* preserve essential services, keep the process within acceptable bounds, and restore authorized operation without introducing a second loss.
- *Anomaly Detection* and *Commensurate Response* connect trustworthy observation to timely containment. Telemetry used for response should be protected from the same compromise it is expected to reveal.

Protection mechanisms that enforce critical constraints should be non-bypassable, evaluable, always invoked, and tamper-resistant [2]. These criteria add valuable questions to Sections 4.2 and 5: Can an alternate protocol, local panel, maintenance port, wireless path, or failover mode bypass enforcement? Is the mechanism simple enough to analyze? Does it mediate every applicable transaction? Can less trusted software alter its policy, code, time, or evidence?

8. Integrate Systems Security Engineering Across the OT Life Cycle

Trustworthy secure design is not a front-end exercise completed when architecture is approved. The protection strategy can be weakened by component substitution, firmware updates, vendor changes, added analytics, revised operating procedures, temporary connections, or accumulated workarounds. Systems security engineering therefore persists through concept, development, acquisition, integration, operation, maintenance, modernization, and disposal.

SP 800-82, Revision 4 can reinforce this life-cycle discipline by connecting its recommendations to recurring systems engineering decisions. *Business and Mission Analysis* identifies critical services and unacceptable losses. *Stakeholder Needs and Requirements Definition* and *System Requirements Definition* translate them

into protection needs and measurable constraints. *Architecture Definition* and *Design Definition* apply SecDOP and security design principles. *Implementation* and *Integration* preserve trusted boundaries and assumptions. *Verification* produces evidence that requirements and mechanisms are satisfied. *Validation* demonstrates that the system supports mission and stakeholder needs in realistic adversity. *Transition*, *Operation*, *Maintenance*, and *Disposal* monitor assumptions, manage changes, rehearse recovery, and protect residual information and capability [2].

This integration is particularly useful for legacy OT. Organizations may not be able to redesign the entire system, but they can still identify the highest-consequence loss scenarios, expose hidden dependencies, reduce unnecessary connectivity, isolate trusted functions, add independent observation, constrain maintenance authority, and build an evidence-backed modernization roadmap. The methodology scales to the available engineering capability.

9. Section-Specific Enhancements to SP 800-82, Revision 4

The following enhancements could be incorporated as short subsections, engineering considerations, callouts, or appendix material. They preserve the publication's risk-based and technology-neutral character while making the path from guidance to trustworthy system design more explicit.

Because Section 4 of the draft is organized around the CSF 2.0 functions, that framework also provides the organizing structure for the *Identify*, *Protect*, *Detect*, *Respond*, and *Recover* enhancements summarized below [7].

Table 2. Actionable Enhancements by SP 800-82 Location

Location	Recommended augmentation	Engineering product
Sec. 3 Risk management and governance	Make unacceptable loss, critical service, system state, and protection need explicit inputs to risk decisions and trade studies.	Loss scenarios; stakeholder protection needs; decision rationale.
Sec. 4.1 Identify	Extend inventories with mission function, trust, authority, dependency, and consequence relationships.	Mission-to-asset map; dependency model; trust-boundary model.
Sec. 4.2 Protect	Apply SecDOP before selecting features; state which security design principles the architecture realizes.	Design rationale; privilege and flow constraints; bypass analysis.
Sec. 4.3 Detect	Specify observability, fidelity, integrity, provenance, independence, and response-time needs.	Detection claims; protected telemetry paths; latency analysis.
Sec. 4.4 Respond	Design bounded, commensurate responses that cannot create a more severe process hazard.	Response constraints; authority model; response validation.
Sec. 4.5 Recover	Define authorized states, trusted recovery paths, minimum essential service, and re-entry criteria.	Recovery architecture; known-good baselines; exercise evidence.
Sec. 5 Architecture	Add emergent behavior, common-mode dependency, trusted-function isolation, and compromise containment analysis.	Assurance-focused architecture views; failure and propagation analysis.
Appendices E and F	Connect control tailoring and RMF artifacts to system claims, evidence, uncertainty, and assurance objectives.	Claim-evidence map; assurance case; residual uncertainty record.

These additions would also improve traceability. A reader could follow a chain from unacceptable loss to system requirement, architectural decision, control implementation, verification method, evidence, residual uncertainty, and risk decision. Such traceability helps operators understand why a safeguard exists and helps change authorities recognize when a modification invalidates an earlier assumption.

10. A Practical Method for Trustworthy Secure OT

Organizations can apply the combined guidance through an eight-step method. The steps are iterative; new evidence, operational experience, incidents, or design changes may require earlier decisions to be revisited.

1. **Frame the mission and system.** Define essential services, stakeholders, operating environments, system boundaries, life-cycle stages, and dependencies. Include cyber, physical, human, and enabling systems.
2. **Define unacceptable losses.** State concrete outcomes that stakeholders cannot tolerate. Relate each loss to affected missions, people, environment, equipment, service, or information.
3. **Model behavior and loss scenarios.** Identify intended functions, critical control actions, system states, initiating conditions, timing constraints, and paths by which failures or adversary actions could produce loss.
4. **Establish protection needs and claims.** Translate the scenarios into system-level requirements and claims. Define the necessary security, safety, resilience, reliability, and performance properties without prematurely selecting a product or control.
5. **Apply SecDOP and design principles.** Eliminate unnecessary capability and connectivity; alter the architecture to reduce propagation and severity; then select mechanisms, monitoring, procedures, and training. Record the rationale and tradeoffs.
6. **Engineer for compromise and degraded operation.** Assume selected components, credentials, or suppliers may be compromised. Bound authority and blast radius; protect essential functions; define safe or acceptable degraded modes; and maintain a trustworthy recovery path.
7. **Build the assurance argument.** For each critical claim, identify evidence and the methods used to obtain it. Address mechanism bypass, tampering, common-mode failure, timing, uncertainty, and the limits of testing.
8. **Continuously preserve trustworthiness.** Monitor assumptions and margins; assess changes; maintain provenance and baselines; rehearse response and recovery; and update the argument when the mission, system, supplier, threat, or operating context changes.

The objective is not a perfect system. It is justified confidence that foreseeable failures and compromises cannot readily become unacceptable operational loss.

11. Evidence and Application: From the SunRISE Pilot to Operational Technology

The value of trustworthy secure systems engineering is supported by preliminary evidence from the NASA/JPL SunRISE pilot project [8]. The pilot compared two digital twins of the SunRISE ground data system. Twin A reflected the traditional cybersecurity and Risk Management Framework approach, while Twin B incorporated selected security design principles from NIST SP 800-160 into the systems engineering life cycle.

When a data-tampering attack was introduced into both twins, the reported mean time to detection decreased from weeks in Twin A to minutes in Twin B. The pilot also found that protections incorporated during design could be more targeted, effective, and economical than safeguards added after system development. Because the results reported in the article were preliminary, they should be treated as

evidence supporting the engineering approach rather than as a definitive comparative evaluation [8]. The SunRISE project was sufficiently successful that NASA decided to conduct a second systems security engineering pilot involving the SPHEREx mission [9].

SunRISE is a space-flight mission rather than an industrial control system. Its engineering lessons nevertheless transfer directly to OT. Both environments support mission-critical operations, impose stringent availability and performance requirements, integrate cyber behavior with consequential operational effects, depend on specialized operators, and may provide limited opportunities for modification after deployment. In both cases, security must be integrated with architecture, mission workflows, anomaly resolution, and life-cycle engineering.

Consider how the same approach could be applied to remote maintenance at a water-treatment facility. SP 800-82 would appropriately lead the facility to inventory the connection, authenticate users, restrict and monitor access, segment networks, plan incident response, and maintain backups. Trustworthy secure design would connect those necessary practices to unacceptable losses such as unsafe chemical dosing, inability to stop dosing, loss of trustworthy process indication, or prolonged interruption of treated-water service.

Applying SecDOP, the facility could eliminate always-on vendor connectivity and unused controller services; terminate remote sessions in a separate maintenance domain; stage changes offline; and require local concurrence before activating revised logic. Independently protected safety limits, out-of-band process indication, separately administered recovery resources, and telemetry that the maintenance session cannot modify would further bound the consequences of compromise.

The facility could then state and verify system-level claims: a remote session cannot directly write active control logic; no single remote identity can authorize activation; loss of an enterprise identity service cannot prevent local safe operation; compromise of the process historian cannot provide a pathway to modify control logic, safety logic, control set points, or active process commands; and a known-good controller configuration can be restored within the maximum tolerable service interruption. Architecture review, configuration inspection, bypass and fault-injection testing, restoration exercises, and supplier evidence would support those claims. The controls remain, but their purpose, placement, and adequacy are tied to unacceptable loss and justified by evidence.

12. Assurance: From Implementation to Justified Confidence

Assurance is the bridge between good intent and trustworthiness. It provides grounds for justified confidence that system claims are or will be achieved [2]. For OT, assurance must address not only functional correctness but also timing, physical effects, operator interaction, supply-chain provenance, failure behavior, environmental conditions, and recovery.

A lightweight *assurance case* can be sufficient. It should state the critical claim, identify assumptions, decompose the claim where necessary, reference objective evidence, record limitations and residual uncertainty, and identify the accountable decision-maker. Evidence can include requirements traceability, system architecture and design analysis, threat and loss-scenario analysis, formal methods for selected mechanisms, code or configuration review, component evaluation, penetration and bypass testing, fault injection, safety-security co-analysis, simulation, operational exercises, and observed performance.

Rigor should be commensurate with consequence, uncertainty, and adversary capability. Not every field device needs formal verification. However, a mechanism that is the sole barrier preventing a catastrophic control action deserves stronger evidence than a low-consequence administrative service. Similarly, a

recovery path that has never been exercised should not be treated as equivalent to one demonstrated under realistic conditions.

This perspective can strengthen the OT overlay and RMF material in Appendices E and F. Control assessment results become part of a broader argument rather than the endpoint. The authorization decision can consider whether the system-level claims are supported, whether important assumptions remain valid, where uncertainty is concentrated, and what operational monitoring is necessary to preserve confidence.

13. Recommended Enhancements to SP 800-82, Revision 4

NIST could strengthen SP 800-82, Revision 4 with a limited set of focused systems security engineering additions that would help ensure the document provides a comprehensive view of OT system protection:

- Add a short systems security engineering subsection in Section 3 that connects operational consequences to unacceptable loss, protection needs, requirements, architecture, evidence, and life-cycle assurance.
- Introduce SecDOP in Section 5 as a prioritization method for OT architecture and compensating-control decisions.
- Add a compact set of OT-focused security design principles and engineering questions, cross-referenced to SP 800-160, Volume 1 rather than reproduced in full.
- Require consideration of emergent behavior, common-mode dependencies, maintenance and recovery paths, and protection-mechanism bypass in architecture reviews.
- Extend the Cybersecurity Framework *Detect*, *Respond*, and *Recover* guidance with trustworthy observation, commensurate response, protective failure, safe degraded operation, authorized recovery states, and evidence requirements.
- Add a claim-evidence-assumption template to an appendix and connect it to control assessment and RMF authorization artifacts.
- Include one worked example showing how an OT loss scenario flows through SecDOP, design principles, controls, verification, and residual-risk acceptance.

These actions are technology-neutral, scalable, and compatible with the Cybersecurity Framework and Risk Management Framework. They would help readers use the publication not only to select and operate safeguards, but also to reason about whether the system architecture makes those safeguards effective.

14. Conclusion

Draft NIST SP 800-82, Revision 4 is an essential source of OT cybersecurity guidance. It recognizes the distinctive consequences, constraints, technologies, and operational realities of cyber-physical systems and provides the practices organizations need to manage risk. Its value should be preserved and amplified.

Trustworthy secure systems concepts provide that amplification. Beginning with mission and unacceptable loss focuses the engineering problem. Treating security as an emergent property of an engineered system addresses interactions and composition. SecDOP prioritizes structural risk reduction before operational compensation. Security design principles guide architecture. *Engineering for Compromise* bounds the consequences of inevitable failures and intrusions. Systems security engineering carries protection through the life cycle. Assurance connects claims to objective evidence and makes residual uncertainty visible.

The next step for OT cybersecurity is not to choose between controls and engineering. It is to use controls within an architecture deliberately engineered to remain trustworthy secure in adversity.

By adding these concepts, principles, and methods, SP 800-82 can help organizations move from protecting OT components to engineering trustworthy secure OT systems—systems that continue to deliver essential services, prevent unacceptable loss, and recover with confidence even when the environment is contested and imperfect.

References

- [1] National Institute of Standards and Technology, *Guide to Operational Technology (OT) Security*, NIST SP 800-82 Rev. 4, Initial Public Draft, September 2026. <https://csrc.nist.gov/pubs/sp/800/82/r4/ipd>
- [2] National Institute of Standards and Technology, *Engineering Trustworthy Secure Systems*, NIST SP 800-160 Vol. 1 Rev. 1, November 2022. <https://csrc.nist.gov/pubs/sp/800/160/v1/r1/final>
- [3] National Institute of Standards and Technology, *Developing Cyber-Resilient Systems: A Systems Security Engineering Approach*, NIST SP 800-160 Vol. 2 Rev. 1, December 2021. <https://csrc.nist.gov/pubs/sp/800/160/v2/r1/final>
- [4] Ron Ross, “Engineering for Compromise: A Strategy for Building Trustworthy Secure Systems,” RONROSSECURE, LLC, 2026. <https://ronrossecure.com/library>
- [5] Ron Ross, “Trustworthy Secure Design,” RONROSSECURE, LLC, 2026. <https://ronrossecure.com/library>
- [6] National Institute of Standards and Technology, *Security and Privacy Controls for Information Systems and Organizations*, NIST SP 800-53 Rev. 5, September 2020, updated December 2020. <https://csrc.nist.gov/pubs/sp/800/53/r5/upd1/final>
- [7] National Institute of Standards and Technology, *The NIST Cybersecurity Framework (CSF) 2.0*, NIST CSWP 29, February 2024. <https://www.nist.gov/cyberframework>
- [8] Ron Ross and Kymie Tan, “Protecting Mission Critical Systems: The Need for a Shift in Culture, Strategy, and Process,” INCOSE INSIGHT, Vol. 28, Issue 3, pp. 15-22, July 2025, <https://doi.org/10.1002/inst.12539>.
- [9] Ron Ross and Kymie Tan, “Space System Survivability Against AI-Driven Cyberattacks: Applying Structural Security Design Principles for Mission Resilience,” KAlperShield, July 2026. <https://ronrossecure.com/library>