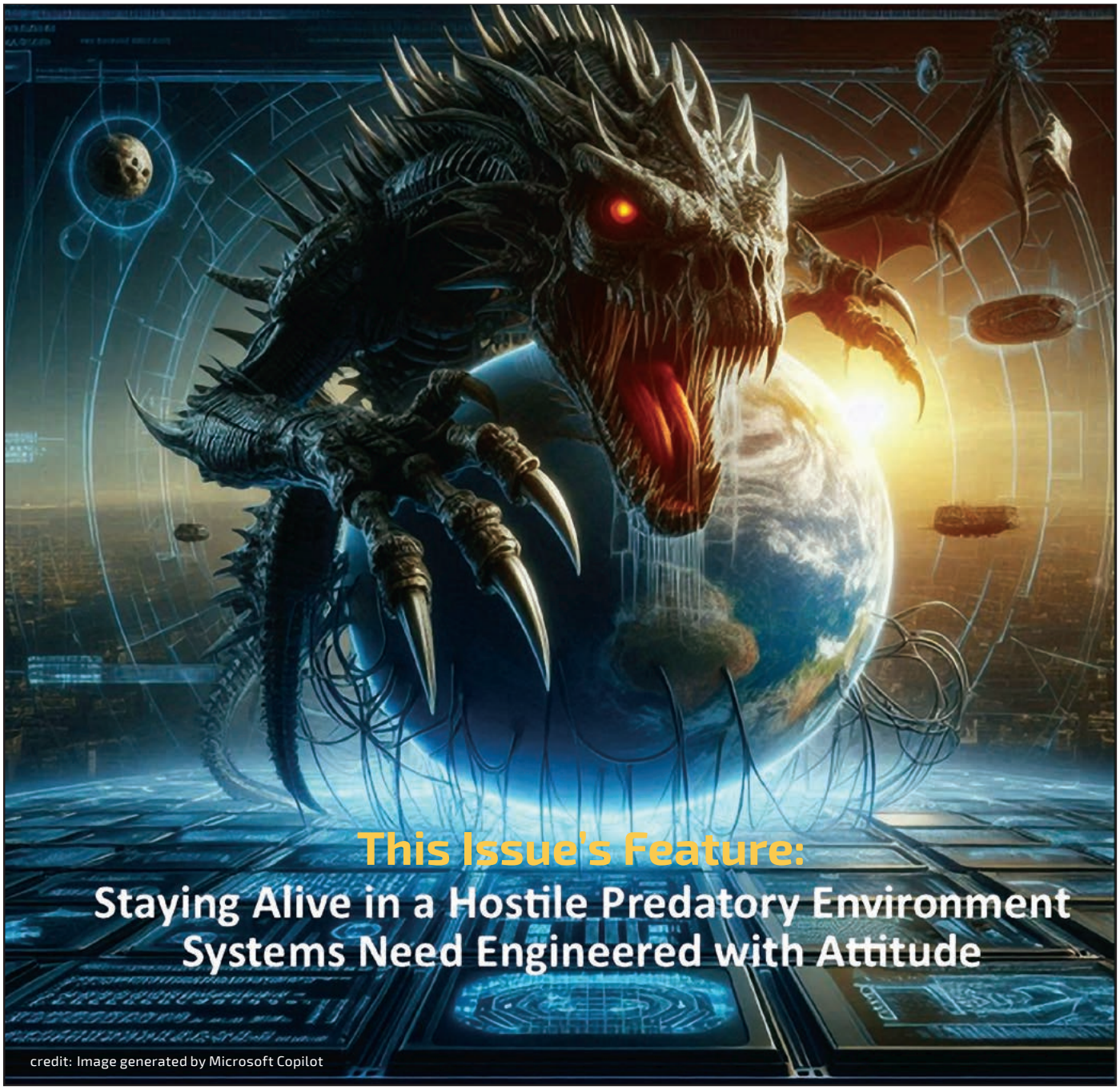
INSIGHT



This Issue's Feature:
Staying Alive in a Hostile Predatory Environment
Systems Need Engineered with Attitude

credit: Image generated by Microsoft Copilot

JULY 2025
VOLUME 28 / ISSUE 3

A PUBLICATION OF THE INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING

Need to Scale Active Learning in the Enterprise?

We Have the Science for That.

Empower your team with the essential digital engineering and machine learning skills to excel in today's competitive landscape. With a strong foundation in SE/MBSE taught by industry-leading experts, your team will be well-equipped to elevate performance and enable data-driven organizational transformation. Custom client programs and public courses available. Explore your options today.

Customizable Learning Programs For Your Organization & Teams

Customizable Programs for Groups and Enterprise Learning

New Advanced Courses in Model-Based Systems Engineering

Programs for Individuals

Short Courses and Certificates for Individuals

Caltech Center for Technology & Management Education

Get started:

ctme.caltech.edu

Connect with an advisor:

execed@caltech.edu

Aerospace • Agriculture • Automotive • Biotech • Chemical • Communications
Defense • Electronics • Energy • Government • High-Tech • Life Sciences
Medical Devices & Diagnostics • Precision Manufacturing • Scientific Research

INSIGHT

A PUBLICATION OF THE INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING

JULY 2025 VOLUME 28/ISSUE 3

Inside this issue

JULY 2025
VOLUME 28/ISSUE 3

About This Publication

INFORMATION ABOUT INCOSE

INCOSE's membership extends to over 25,000 members and CAB associates and more than 200 corporations, government entities, and academic institutions. Its mission is to share, promote, and advance the best of systems engineering from across the globe for the benefit of humanity and the planet. INCOSE charters chapters worldwide, includes a corporate advisory board, and is led by elected officers and directors.

For more information, click here:
The International Council on Systems Engineering
(www.incose.org)
INSIGHT is the magazine of the International Council on Systems Engineering. It is published six times per year and

OVERVIEW

features informative articles dedicated to advancing the state of practice in systems engineering and to close the gap with the state of the art. ***INSIGHT*** delivers practical information on current hot topics, implementations, and best practices, written in applications-driven style. There is an emphasis on practical applications, tutorials, guides, and case studies that result in successful outcomes. Explicitly identified opinion pieces, book reviews, and technology roadmapping complement articles to stimulate advancing the state of practice. ***INSIGHT*** is dedicated to advancing the INCOSE objectives of impactful products and accelerating the transformation of systems engineering to a model-based discipline.
Topics to be covered include resilient systems, model-based systems engineering, commercial-driven transformational systems engineering, natural systems, agile security, systems of systems, and cyber-physical systems across disciplines and domains of interest to the constituent groups in the systems engineering community: industry, government, and academia. Advances in practice often come from lateral connections of information dissemination across disciplines and domains. ***INSIGHT*** will track advances in the state of the art with follow-up, practically written articles to more rapidly disseminate knowledge to stimulate practice throughout the community.

Editor-In-Chief William Miller
insight@incose.net +1 908-759-7110

Layout and Design Chuck Eng
chuck.eng@comcast.net

Member Services INCOSE Administrative Office
info@incose.net +1 858 541-1725

PERMISSIONS

*** PLEASE NOTE: If the links highlighted here do not take you to those web sites, please copy and paste address in your browser.**

Permission to reproduce Wiley journal Content:
Requests to reproduce material from John Wiley & Sons publications are being handled through the RightsLink® automated permissions service.

Simply follow the steps below to obtain permission via the Rightslink® system:

- Locate the article you wish to reproduce on Wiley Online Library (http://onlinelibrary.wiley.com)
- Click on the 'Request Permissions' link, under the ‹ARTICLE TOOLS› menu on the abstract page (also available from Table of Contents or Search Results)
- Follow the online instructions and select your requirements from the drop down options and click on 'quick price' to get a quote
- Create a RightsLink® account to complete your transaction (and pay, where applicable)
- Read and accept our Terms and Conditions and download your license
- For any technical queries please contact customercare@copyright.com
- For further information and to view a Rightslink® demo please visit www.wiley.com and select Rights and Permissions.

AUTHORS – If you wish to reuse your own article (or an amended version of it) in a new publication of which you are the author, editor or co-editor, prior permission is not required (with the usual acknowledgements). However, a formal grant of license can be downloaded free of charge from RightsLink if required.

Photocopying
Teaching institutions with a current paid subscription to the journal may make multiple copies for teaching purposes without charge, provided such copies are not resold or copied. In all other cases, permission should be obtained from a reproduction rights organisation (see below) or directly from RightsLink®.

Copyright Licensing Agency (CLA)
Institutions based in the UK with a valid photocopying and/or digital license with the Copyright Licensing Agency may copy excerpts from Wiley books and journals under the terms of their license. For further information go to CLA.

Copyright Clearance Center (CCC)
Institutions based in the US with a valid photocopying and/or digital license with the Copyright Clearance Center may copy excerpts from Wiley books and journals under the terms of their license, please go to CCC.

Other Territories: Please contact your local reproduction rights organisation. For further information please visit www.wiley.com and select Rights and Permissions.
If you have any questions about the permitted uses of a specific article, please contact us.

Permissions Department – UK
John Wiley & Sons Ltd.
The Atrium,
Southern Gate,
Chichester
West Sussex, PO19 8SQ
UK
Email: Permissions@wiley.com
Fax: 44 (0) 1243 770620
or

Permissions Department – US
John Wiley & Sons Inc.
111 River Street MS 4-02
Hoboken, NJ 07030-5774
USA
Email: Permissions@wiley.com
Fax: (201) 748-6008

ARTICLE SUBMISSION insight@incose.net

Publication Schedule. ***INSIGHT*** is published six times per year. Issue and article submission deadlines are as follows:

- September 2025 issue – 1 June 2025
- October 2025 – 1 July 2025
- December 2025 – 1 September 2025
- February 2026 issue – 1 November 2025
- April 2026 issue – 2 January 2026
- June 2025 issue – 1 March 2026

For further information on submissions and issue themes, visit the INCOSE website: www.incose.org

 ISSN 2156-485X; (print) ISSN 2156-4868 (online)

ADVERTISE

Readership
INSIGHT reaches over 25,000 members and CAB associates and uncounted employees and students of more than 130 CAB organizations worldwide. Readership includes engineers, manufacturers/purchasers, scientists, research and development professionals, presidents and chief executive officers, students, and other professionals in systems engineering.

Issuance	Circulation
2025, Vol 28, 6 Issues	100% Paid

Contact us for Advertising and Corporate Sales Services
We have a complete range of advertising and publishing solutions professionally managed within our global team. From traditional print-based solutions to cutting-edge online technology the Wiley-Blackwell corporate sales service is your connection to minds that matter. For an overview of all our services please browse our site which is located under the Resources section. Contact our corporate sales team today to discuss the range of services available:

- Print advertising for non-US journals
- Email Table of Contents Sponsorship
- Reprints
- Supplement and sponsorship opportunities
- Books
- Custom Projects
- Online advertising

Click on the option below to email your enquiry to your nearest office:

- Asia and Australia corporatesalesaustralia@wiley.com
- Europe, Middle East and Africa (EMEA) corporatesaleseurope@wiley.com
- Japan corporatesalesjapan@wiley.com
- Korea corporatesaleskorea@wiley.com

USA (also Canada, and South/Central America):

- Healthcare Advertising corporatesalesusa@wiley.com
- Science Advertising Ads_sciences@wiley.com
- Reprints Commercialreprints@wiley.com
- Supplements, Sponsorship, Books and Custom Projects busdev@wiley.com

Or please contact: Marcom@incose.net

CONTACT

Questions or comments concerning:

Submissions, Editorial Policy, or Publication Management
Please contact: William Miller, Editor-in-Chief
insight@incose.net

Advertising—*please contact:*
Marcom@incose.net

Member Services – *please contact:* info@incose.org

ADVERTISER INDEX

July Volume 28-3

CORPORATE ADVISORY BOARD — MEMBER COMPANIES

Advanced Systems Engineering, LLC
Aerospace Corporation, The
Airbus
Albers Aerospace
AM General LLC
Analog Devices, Inc.
ANSYS, Inc
Arcfield
Auburn University
Australian National University
AVIAGE SYSTEMS
Aviation Industry Corporation of China, LTD
BAE Systems
Bechtel
Becton Dickinson
Belcan Engineering Group LLC
BMT Canada
Boeing Company, The
Booz Allen Hamilton Inc.
Boston Scientific Corporation
BTS Software Solutions
California State University Dominguez Hills
Capgemini Engineering
Carnegie Mellon University Software Engineering Institute
Change Vision, Inc.
Colorado State University Systems Engineering Programs
Cornell University
Cranfield University
C.S. Draper Laboratory, Inc.
Cubic Corporation
Cummins, Inc.
Cybernet MBSE Co, Ltd
Dassault Systèmes
Defense Acquisition University
Deloitte Consulting, LLC
Denso Create Inc
DENTSU SOKEN INC
Drexel University
Eaton
Eindhoven University of Technology
EMBRAER
FAMU-FSU College of Engineering
Federal Aviation Administration (U.S.)
Florida Institute of Technology
Ford Motor Company
GE Aerospace
General Dynamics
General Motors
George Mason University
Georgia Institute of Technology
Hitachi Energy
Honeywell Aerospace Technologies
Huawei Technologies Co. Ltd
IBM
Idaho National Laboratory
IQNOX, LLC
ISAE - Supaero
ISDEFE
IVECO Group
Jama Software
Jet Propulsion Laboratory
John Deere & Company
Johns Hopkins University
KBR, Inc.
KEIO University
L3Harris Technologies
Lawrence Livermore National Laboratory
Leidos
LEONARDO
Lockheed Martin Corporation
Los Alamos National Laboratory
Loyola Marymount University
Magna
ManTech International Corporation
Marquette University
Massachusetts Institute of Technology
MBDA (UK) Ltd
Medtronic
MetaTech Consulting Inc.
Missouri University of Science & Technology
MITRE Corporation, The
Mitsubishi Electric Corporation
Mitsubishi Heavy Industries, Ltd
Modern Technology Solutions Inc
National Aeronautics and Space Administration (NASA)
National Reconnaissance Office (NRO)
National Security Agency Enterprise Systems
Nissan Motor Co, Ltd
Northrop Grumman Corporation
Pacific Northwest National Laboratory
Pennsylvania State University
Petronas International Corporation Limited
Prime Solutions Group, Inc
Project Performance International (PPI)
Purdue University
QRA Corporation
RealmOne
Rolls-Royce
RTX
Saab AB
SAFRAN
SAIC
Sandia National Laboratories
Saudi Railway Company
SENSEONICS
Shanghai Formal-Tech Information Technology Co., Ltd
Shell
Siemens
Sierra Nevada Corporation
Singapore Institute of Technology
Southern Methodist University
SPEC Innovations
Stevens Institute of Technology
Strategic Technical Services LLC
Studio SE, Ltd.
Swedish Defence Materiel Administration (FMV)
Systems Planning and Analysis
Taiwan Space Agency
Tata Consultancy Services
Thales
The George Washington University
The University of Arizona
The University of Utah
Torch Technologies
TOSHIBA Corporation
Trane Technologies
Tsinghua University
UK MoD
UNCOMN
Universidade Federal De Minas Gerais
University of Alabama in Huntsville
University of Arkansas
University of California San Diego
University of Connecticut
University of Maryland
University of Maryland, Baltimore County
University of Maryland Global Campus
University of Michigan, Ann Arbor
University of New South Wales, The, Canberra
University of South Alabama
University of South-Eastern Norway (USN)
University of Texas at El Paso (UTEP)
US Department of Defense
Veoneer US Safety Systems, LLC
Virginia Tech
Volvo Cars Corporation
Volvo Construction Equipment
Wabtec Corporation
Wayne State University
Weber State University
Wichita State University College of Engineering
Woodward Inc
Worcester Polytechnic Institute (WPI)
Woven by Toyota, Inc.
Zuken, Inc

FROM THE EDITOR-IN-CHIEF

William Miller, insight@incose.net

We are pleased to announce the July 2025 *INSIGHT* issue published cooperatively with John Wiley & Sons as the systems engineering practitioners' magazine. The *INSIGHT* mission is to provide informative articles on advancing the practice of systems engineering and to close the gap between practice and the state of the art as advanced by *Systems Engineering*, the Journal of INCOSE also published by Wiley. The theme of this issue is illuminated by the bold cover statement: *Staying Alive in a Hostile Predatory Environment – Systems Need Engineered with Attitude*. We thank theme editor Rick Dove and the authors for their contributions. Several key takeaways of this issue are: 1) "security throughout the system lifecycle as foundational a perspective in systems design as system performance and safety are today" and not an add-on; 2) loss-driven engineering approach; 3) from non-functional security requirements to functional requirements; and 4) applying and validating the NIST (US National Institute of Standards and Technology) Risk Management Framework (RMF).

The future of systems engineering (FuSE) is to realize the *Systems Engineering Vision 2035*, freely accessible at https://www.incose.org/about-systems-engineering/se-vision-2035. FuSE began in late 2017 leveraging the previous *Systems Engineering Vision 2025* and in anticipation of the latest vision announced at the 2022 International Workshop in January 2022. FuSE has identified four streams to drive implementation to realize the Vision 2035: systems engineering vision & roadmap, systems engineering foundations, systems engineering methodology, and systems engineering application extensions. Security has been central to FuSE and a recurrent *INSIGHT* theme with agile system-security: sustainable systems evolve with their environment (July 2016), cyber secure and resilient approaches with feature-based product line engineering (September 2020) and setting the current context for security in the Future of Systems Engineering (June 2022).

Rick Dove leads off the July 2025 *INSIGHT* with "Attitudes," noting that digital controls and Internet connectivity have unintended consequences resulting in a hostile predatory environment with organized crime and nation state interests exploiting readily accessible opportunities for financial and political advantage. This has brought a new dimension to security concerns – the traditional focus on information protection must now also contend with proactive functional protection. Dove describes systems with attitude having perpetual security systems: an engine of vigilance and an engine of resilience.

"Using Systems Thinking to Advance Security in the Future of Systems Engineering (FuSE), a Progress Report," by the Systems Security Working Group describes a needs-oriented, loss-driven, capability-based analysis to define security strategies that become functional requirements that promotes stakeholder alignment of the security vision and leads to effective security tactics and techniques that collectively achieve the security strategies. The path is to transition practices to a future where our systems are designed to achieve and sustain security as an intentional capability of the system throughout its lifecycle resulting is a system that achieves functional perseverance in a hostile predatory environment.

"Protecting Mission Critical Systems: The Need for a Shift in Culture, Strategy, and Process," by Ron Ross and Kymie Tan describe insights in contrasting the traditional compliance-based approach to protecting space systems to that using the NIST Risk Management Framework (RMF), a trustworthy secure systems engineering approach as described in the NIST Special Publication 800-160. A change in strategy and approach from the traditional way is necessary to address the modern sophisticated cyber adversary operating in a world of highly complex and evolving systems.

"Guide to Security Needs and Requirements – Making Security a Functional Requirement" by Beth Wilson describes the joint project of the Systems Security Working Group (SSWG) and Requirements Working Group (RWG) to create a *Guide to Security Needs and Requirements* targeting both the systems engineering practitioner and the systems security practitioner to help them collaboratively define security needs and requirements that result in a secure system in operation. The approach to

perform needs-oriented, loss-driven, capability-based analysis across the systems engineering activities resulting in a set of need statements capturing the stakeholder expectations concerning security and a set of functional requirements defining what the system must do to address those needs. Defining security as a functional requirement helps design a system that can prepare for, defend against, and recover from adversity to achieve and sustain mission success.

"Governance and Resilience: A Holistic Approach to Systems Security in Complex and Chaotic Environments" by Sue Caskey and Adam Williams highlights the challenges faced by nuclear power plant operators in predatory contexts and the importance of integrating security objectives into governance frameworks. By incorporating security as a fundamental aspect of governance, the article underscores its significance for persistence, adaptation, and transformation in the face of uncertainty. Additionally, the authors introduce key heuristics of systems security, such as the importance of context, knowledge-based decision-making, and organization-specific sociological factors.

"A Model-Based Approach for Privacy Risk Mitigation Integrating Systems Engineering with System-Theoretic Process Analysis" by David Hetherington examines the design of a digital personal communications device designed to achieve security goals and demonstrate the use of system-theoretic process analysis (STPA) in the analysis of a proposed design. David describes a model-based approach to the design work which represents the recently released standard SAE J3307 "System Theoretic Process Analysis (STPA) Standard for All Industries" (*J3307_202503, 2025*) which specifies an auditable workflow for the STPA methodology originally described in the STPA Handbook.

"How Security Needs Systems Engineering" by Mark Winstead addresses security as a system problem, where the security engineering must not be done stove piped from system engineering. The discussion addresses the role of systems thinking and the need for evidence-based assurance overseen by systems engineering.

"Illuminating Systems Security Through Case Studies – Much More than Controls" by Beth Wilson reviews cyber-attack case studies to examine security challenges and failures holistically using systems thinking, considering technical concerns, business decisions, and human behaviors that made the attack possible, and explore systems security concepts from a systems engineering perspective.

"When Malicious Actors Control Your Subsystems: A Systems Engineering Approach to Functional Perseverance" by David Hetherington and Ivan Taylor examines a method for using system-theoretic process analysis (STPA) and system dynamics (SD) to enhance security-aware system engineering. Security in modern engineered systems is not merely an added layer of protection but a prerequisite for system functionality. As systems engineers navigate the evolving security landscape, they must prioritize functional perseverance, the ability of a system to maintain operational integrity despite adversarial threats.

"AI for System Security Design: A Good Tool or a Dangerous Weapon?" by Beth Wilson warns of the temptation for systems engineers to use AI tools to quickly generate security requirements and skip engagement with systems security practitioners. The proliferation of AI tools that have been trained with security controls invites misguided approaches that deliver systems that are not secure in the operational environment. AI literacy is important to understand both the benefits and the limitations of AI to use it ethically and effectively.

We hope you find *INSIGHT*, the practitioners' magazine for systems engineers, informative and relevant. Feedback from readers is critical to *INSIGHT*'s quality. We encourage letters to the editor at insight@incose.net. Please include "letter to the editor" in the subject line. *INSIGHT* also continues to solicit special features, standalone articles, book reviews, and op-eds. For information about *INSIGHT*, including upcoming issues, see https://www.incose.org/publications/insight. For information about sponsoring *INSIGHT*, please contact the INCOSE marketing and communications director at marcom@incose.net. ■

INSIGHT Special Feature

Attitudes

Rick Dove, dove@parshift.com

■ **ABSTRACT**
Digital controls and internet connectivity have fostered a hostile predatory environment for modern systems. Organized crime and nation state interests are naturally compelled to exploit these readily accessible opportunities for financial and political advantages. Systems engineering is being called upon to reorient its priorities accordingly. INCOSE's Future of Systems Engineering (FuSE) to realize the Systems Engineering Vision 2035 has a security-focused activity exploring what this reorientation might be. This article shares some of that thinking, exposes some issues in need of more thinking, and suggests why and how all systems engineers could and should be part of this thinking.

INTRODUCTION

Predatory hostility is now an active characterization of a system's operational environment, with damage, disruption, and extortion the intended outcomes.

Originally conceived in the industrial age, systems engineering faces different demands in the digital age. A key shift occurred when manual and mechanical system controls transitioned to digital controls with network accessibility. This controls shift brought a new dimension to security concerns—the traditional focus on information protection now has to contend with functional protection as well.

A few touch points illuminate the concern:

- *Systems Engineering Vision 2035* published by INCOSE imagines a future with "security as foundational a perspective in systems design as system performance and safety are today" (INCOSE 2021, p.37).
- The US Defense Department's Chief Information Officer "plans on 'blowing up' outdated software risk framework" (Welch 2025).
- Microsoft's Executive Vice President says "We are making security our top priority at Microsoft, above all else" (Bell 2024).
- NASA/JPL just completed a revealing experiment comparing security systems engineering a la NIST 800-160 with its traditional systems security engineering (Ross and Tan 2025).

The points above are about getting a broader base of people involved in the security systems engineering equation.

INCOSE's Systems Security working group, a mixed composition of systems security engineers and systems engineers, is exploring the role of systems engineering and systems engineers in the security systems engineering equation – with some discomfort. It doesn't appear to be a problem with a straightforward engineering answer. This article will share the sources of some of that discomfort before concluding with how systems engineers might play a part in the solution. The purpose of this article is to instigate some thinking and participation in finding an effective path.

RELATIONSHIPS

In working on the security aspects of the Future of Systems Engineering (FuSE) to realize the *Systems Engineering Vision 2035* the security working group developed a roadmap (Dove et al. 2021) that positioned *Security Proficiency in the Systems Engineering Team* as a central strategic concept. In wrestling with that concept it became evident that the original roadmap thoughts of embedding professional security expertise within the systems engineering team were unrealistic. Professional security engineers with appropriate systems breadth are in scarce supply relative to the demand. A different approach emerged based on systems engineers doing what systems engineers already know how to do—with an explicit focus on enabling and facilitating the needs of security systems engineers (Dove et al. 2025).

NIST 800-160 (Ross, Winstead, and McEvilley 2022, p.29) describes systems security engineering as a subdiscipline of systems engineering. For discussion

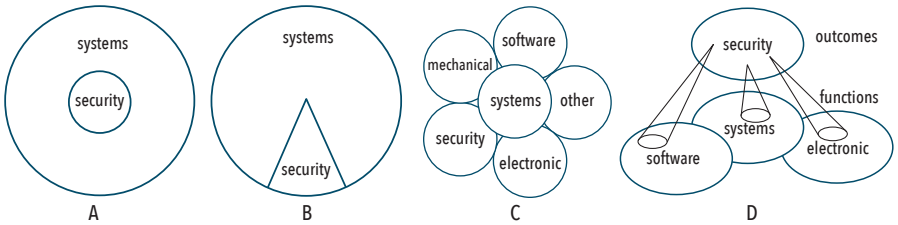


Figure 1. Relationship options

purposes an attempt was made to graphically depict this relationship as the A and B options in Figure 1.

Initial interest in the A and B options didn't survive long in discussion among working group members, as some indicated strong incompatibility with the culture and views of their organizations. Option C was preferred over A and B as something more acceptable, but didn't ring as a comfortable expression of security's involvement with all the engineering disciplines. That discomfort resulted in the option D depiction, which exhibited some intellectual resonance but didn't speak to systems engineering in the intended sense.

Any of these depictions might find a cultural fit somewhere, and collectively they might prompt a thoughtful exploration of the issue. Systems engineering and the organizations that employ systems engineering cannot ignore this issue. As the saying goes, it is the elephant in the room: how should systems engineering and security engineering relate to each other in a mutually synergistic cooperative endeavor?

Bottom line, the working group found the alternate depiction discussion enlightening, and convincing that the hierarchy of needs depicted in Figure 2 (Dove et al. 2025) is a most appropriate representation of security's relationship to systems engineering.

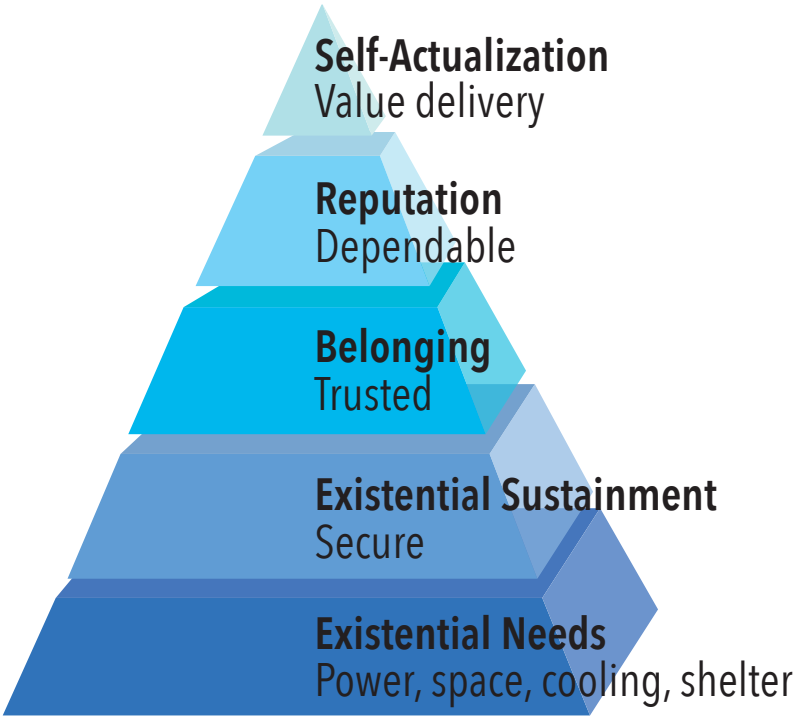


Figure 2. Technical hierarchy of needs – staying alive is a prerequisite of system functionality (adaptation of Maslow's Hierarchy)

Figure 3. Metaphorically, like chocolate in cake, security can be in spots or infused

DISCONNECTS

Attempting to counter the after-system-design-is-done engagement of systems security engineering activity, a call "to bake security in" has been echoed about in recent years. The bake-it-in metaphor is a meaningful contrast to icing on the top but leaves the baking methods open for question.

Two thoughts from working group discussions moves this thinking forward:

- Adding chocolate to a cake doesn't make it a chocolate cake anymore than adding security to a system makes it a secure system (Figure 3).
- Conway's Law (loosely): organizations design systems which reflect the organization.

There is a growing dis-ease with the increasing toll of hostile predatory activity. The 2021 FuSE roadmap for improving systems security identified eleven concepts ready for timely improvement. But considering Conway's Law these concepts look more like attending to symptoms rather than causes.

We (in the large) are uneasy about the situation and are seeing various thoughtful sense-making proposals for approaches that will provide relief; for example, education of systems engineers in security knowledge, principles for secure design, early involvement of system security engineering with systems engineering activity, and other such. But these could be viewed as addressing the symptoms of the dis-ease.

Conway's Law offers a conceptual pattern that explains why we don't have an infused security cake. The respect, appreciation, and necessity for security is not a front-burner organizational issue. For a publicly notable contrast, complete with full infusion details, see Microsoft's May 2024 declaration "… security our top priority at Microsoft, above all else" (Bell 2024), with periodic progress reports as proof of life for the cynics (Bell 2025).

ATTITUDES

A recent working group paper (Dove et al. 2025) explored the role of systems engineering in creating perseverant systems — ones designed to endure and prevail in an environment of constantly evolving, intelligently-directed, predatory hostility. The objective was to give systems engineers an embraceable role in the systems security equation. The approach offered a simple mental model of what should be done for who and why and showed that systems engineering skills are adequate for the job. Perseverance is a systems engineering design task that moves the security issue up front and central to

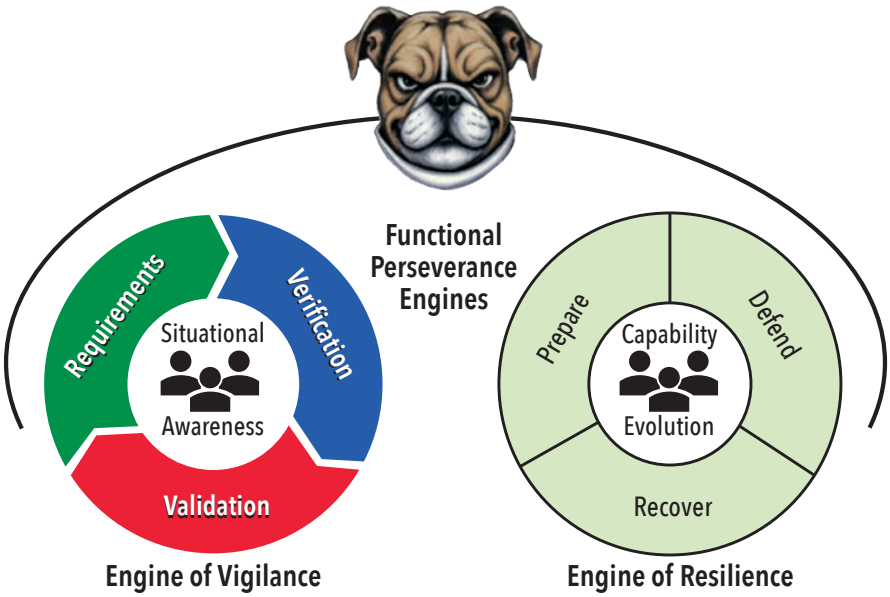


Figure 4. Systems with attitude have perpetual security systems engineering

the system engineering activity.

The model in Figure 4 represents two engines responsible for perpetual situational awareness and capability evolution relative to a system's security needs. Systems engineering leads the Engine of Vigilance activities while security engineering leads the Engine of Resilience activities—with engines working together like cog-wheeled gears and engineers working together like synergistic partners.

The bulldog represents a system's functional attitude of perpetual self preservation.

Systems with attitude need systems engineering with attitude. Shamelessly adopted from a 1976 movie script (*Network* 1976):

> *Go to your windows. Open them and stick your head out and yell – 'I'm as mad as hell and I'm not gonna take this anymore!' Things have got to change. But first, you've gotta get mad! …You've got to say, 'I'm as mad as hell, and I'm not gonna take this anymore!' Then we'll figure out what to do.*
>
> Sidney Aaron "Paddy" Chayefsky
> *Network* (1976 movie)
> https://www.youtube.com/watch?v=QMBZDwf9dok

Internet connected systems are free game in a hostile predatory environment. Not just those that are designed, built and sold, but also the ones that support the abilities to design, build, and sell. It's time for systems engineers and systems engineering to do something about that—directly in the work at hand, and indirectly in helping the larger at-work community understand what systems thinking reveals. ■

Figure 5. I'm not gonna take this anymore!

ACKNOWLEDGEMENTS

The thoughts expressed in this article emerged from biweekly workshops held throughout 2024 and 2025 by a team of collaborators. Alphabetically: Rick Dove, Mona Humes, Greg Leach, Richard Massey, Gerry Ourada, Barry Papke, Adam Scheuer, Martin Span Daniel Sudmeier, Luke Thomas, Adam Williams, Beth Wilson, and Mark Winstead.

Some graphics in in this article were created by generative AI applications. Figure 3 emerged when xAI's Grok was asked to draw a slice of cake that has some swirls of chocolate in it. … then … replace the white cake with chocolate colored cake. The bulldog head in Figure 4 was Grok's image for a mean bulldog head. Figure 5 emerged when deepai.org's AI image generator was asked to create an image of an angry man yelling out of a window.

REFERENCES

- Bell, C. 2024. "Security above all else—expanding Microsoft's Secure Future Initiative." Microsoft, 3 May. www.microsoft.com/en-us/security/blog/2024/05/03/security-above-all-else-expanding-microsofts-secure-future-initiative.
- Bell, C. 2025. "Securing our future: April 2025 progress report on Microsoft's Secure Future Initiative." Microsoft, 21 April. www.microsoft.com/en-us/security/blog/2025/04/21/securing-our-future-april-2025-progress-report-on-microsofts-secure-future-initiative/.
- Dove, R., K. Willett, T. McDermott, H. Dunlap, D. P. MacNamara, and C. Ocker. 2021. "Security in the Future of Systems Engineering (FuSE), a Roadmap of Foundational Concepts." INCOSE International Symposium, Virtual, 17-22 July.
- Dove, R., M. Humes, G. Leach, R. Massey, G. Ourada, B. Papke, A. Scheuer, M. Span, D. Sudmeier, L. Thomas, A. Williams, B. Wilson, and M. Winstead. 2025. "Systems Engineering with Attitude." INCOSE International Symposium, Ottawa, CA, 26-31 July.
- House Committee on Homeland Security. 2024. Cyber Threat Snapshot. U.S. House of Representatives. 12-November. https://homeland.house.gov/wp-content/uploads/2024/11/11.12.24-Cyber-Threat-Snapshot.pdf.
- INCOSE. 2021. *Systems Engineering Vision 2035*. International Council on Systems Engineering.
- Microsoft. 2025. "Secure Future Initiative (SFI) Security above all else." April.
- Ross, R., M. Winstead, and M. McEvilley. 2022. Engineering Trustworthy Secure Systems. Special Publication NIST 800-160v1r1 November. https://doi.org/10.6028/NIST.SP.800-160v1r1.
- Ross, R., and K. Tan. 2025. "Protecting Mission Critical Systems – The Need for a Shift in Culture, Strategy, and Process." *INSIGHT* 28 (3), July.
- Welch, C. 2025. "How a key Pentagon tech leader plans on 'blowing up' outdated software risk framework." Breaking Defense. 9 May. https://breakingdefense.com/2025/05/how-a-key-pentagon-tech-leader-plans-on-blowing-up-outdated-software-risk-framework/.

ABOUT THE AUTHOR

Rick Dove is unaffiliated, an INCOSE Fellow, chairs INCOSE's System Security Engineering working group, and leads INCOSE's Future of Systems Engineering (FuSE) Security project.

Using Systems Thinking to Advance Security in the Future of Systems Engineering (FuSE), a Progress Report

Systems Security Working Group, systemssecurityengineering@incose.net

■ **ABSTRACT**
The Security in the Future of Systems Engineering (FuSE) team has made significant progress since its launch to realize the INCOSE vision described in *Systems Engineering Vision 2035* (INCOSE 2021). The output products to date promote improved systems engineering practices to achieve security as a foundational perspective. The systems thinkers on this team have performed holistic analysis of current practices to expose existing anti-patterns and mental models that informed the transformation to future practices that can yield desired results and achieve the 2035 vision. Needs-oriented, loss-driven, capability-based analysis to define security strategies that become functional requirements promotes stakeholder alignment of the security vision and leads to effective security tactics and techniques that collectively achieve the security strategies. The result is a system that achieves functional perseverance in a hostile predatory environment. The work products completed so far and those in progress reflect our efforts to transition practices to a future where our systems are designed to achieve and sustain security as an intentional capability of the system throughout its lifecycle.

INTRODUCTION

The INCOSE Systems Security Working Group (SSWG) launched a dedicated team to transform systems engineering practices to realize the INCOSE vision described in the *Systems Engineering Vision 2035* (INCOSE 2021) that by 2035 "security will be as foundational a perspective in systems design as system performance and safety are today." This has been, and continues to be, an intensive effort with a core team that meets bi-weekly to explore the current state and develop the future state of systems security. The Security in the Future of Systems Engineering (FuSE) roadmap defines 6 objectives and 11 foundation concepts that support the achievement of these objectives that were published at the INCOSE International Symposium in 2021 (Dove 2021) and summarized in Figure 1.

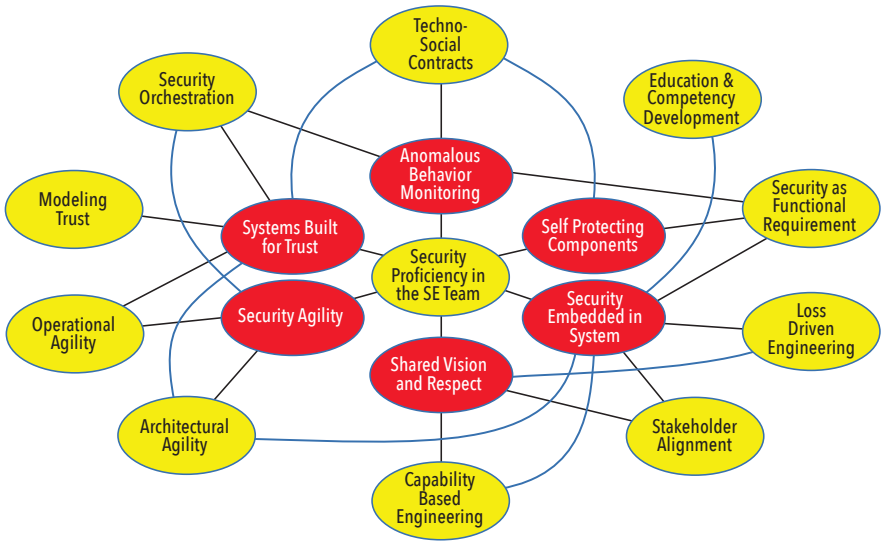


Figure 1. Security in the Future of Systems Engineering (FuSE) foundation concepts (yellow) support the achievement of strategic objectives (red)

Table 1. Work products completed and in progress for the security in FuSE effort

Date Released	Work Product	Summary
July 2021	IS21 Paper: Security in FuSE, a Roadmap of Foundational Concepts	Established foundation concepts
June 2022	*INSIGHT* Special Issue: Security in the Future of Systems Engineering	*INSIGHT* articles addressing foundation concepts
July 2022	IS2022 Tutorial: Systems Security Engineering, a loss-focused approach	Loss-driven analysis
July 2023	IS2023 Paper: Democratizing Systems Security	Stakeholder alignment foundation concept synergy with other foundation concepts
July 2024	IS2024 Tutorial: Security as a Foundational Perspective in Systems Engineering: Engineering Trustworthy Secure Systems	Security proficiency elements of systems engineering aligned with Security in FuSE
August 2024	Guide to Security Needs and Requirements	Needs-oriented, loss-driven, capability-based analysis for functional security requirements
July 2025	IS2025 Paper: Systems Engineering with Attitude	Role of systems engineers in creating perseverance in predatory environments
July 2025	*INSIGHT* Special Issue: Stayin' Alive is Essential – Security is a System Engineer's Problem	*This issue*
In progress	Security Primer for Systems Engineers	What stakeholders (including systems engineers) need to know about system security to achieve functional perseverance
In progress	Systems Security Micro-Credential	Awareness level proficiency in systems security topics for systems engineering

As systems thinkers, the core team began to recognize intersections and interdependencies among the foundation concepts, anti-patterns resulting from these interactions, and mental models influencing the observed behavior related to security in the current systems engineering practices. The INCOSE Systems Engineering Competency Framework includes Systems Thinking as a competency area (INCOSE 2025). In the "why it matters" section of this competency area, the need for systems thinking is described as "understanding how actions and decisions in one area affect another, and that the optimization of a system within its environment does not necessarily come from optimizing the individual system components." Applying systems thinking techniques to advance the state of the art related to systems security means looking at the current and potential future practices holistically. We applied systems thinking techniques to understand "why does what we do now not work as intended?"

The results so far and the activity still in progress reflect our efforts to transition practices to a future where our systems are designed to achieve and sustain security as an intentional capability of the system throughout its lifecycle. Table 1 summarizes these work products.

APPLYING SYSTEMS THINKING TO SECURITY IN FUSE

A key systems thinking tool is the iceberg model shown in Figure 2. Systems thinkers can use the iceberg tool to identify visible events and patterns of behavior, explore the visible patterns to understand the underlying structures, and analyze the interactions in the structures to expose hidden mental models. Other systems thinking tools such as causal loops and dynamic modeling show how the interactions between the structure elements result in the effects that are visible as patterns of behavior. Systemic root cause analysis breaks the feedback loops in the anti-patterns and reveals new patterns that can produce the desired behaviors.

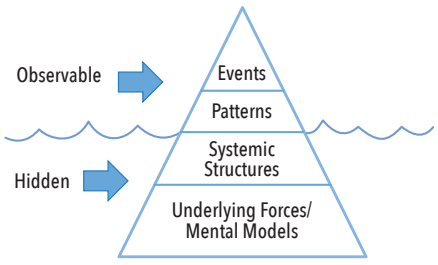


Figure 2. The iceberg model is a systems thinking tool where we identify the hidden mental models and systemic structures that give rise to events and patterns that can be observed (from Monat 2017)

The team identified visible behavior and identified general problems to solve that comprised the Security in FuSE roadmap (Dove 2021 and Dove 2022). Some of these visible patterns included:

- Miscommunication exists across knowledge and expertise boundaries.
- Misalignment of security vision among stakeholders leads to poor security outcomes.
- Security tactics selected based on available solutions do not yield desired results.
- Vulnerability assessments occur after design is complete.
- Disparate security solutions operate independently and ineffectively.
- Systems are slow to detect, respond to, and recover from an attack.
- Systems are unable to respond to innovative threats and attacks.

As we explored these observable patterns, we used the 11 foundation concepts to represent an underlying structure. In trying to understand why it is so difficult for stakeholders to articulate security needs,

we recognized that systems security is more than a collection of techniques and people with expertise to implement these techniques. Demanding that security be a consideration earlier in the system design activities only serves to accelerate instances where systems engineers abdicate responsibility for security relegating security specialists to an external silo of activity. We recognized that systems security is a mission that needs an aligned team of stakeholders. To achieve that alignment, we must have understandable and achievable security capabilities. Using the definition of democratization as "the action of making something accessible to everyone" we saw stakeholder alignment as synergistic with all the foundation concepts resulting in the paper "Democratizing Systems Security" (Dove 2023).

As we continued to explore the dynamic relationships among the foundation concepts, we identified patterns of behavior that captured how current approaches fail to produce secure systems. We identified hidden mental models responsible for observable behaviors including:

- Stakeholders (including systems engineers) offload responsibility for security outcomes.
- Security is a set of non-functional requirements that are allocated to security practitioners to implement.
- Security requirements are defined by selecting security controls from NIST 800-53 (NIST 2020).
- Security verification then becomes compliance-driven confirmation of the selected controls.
- Security focus is limited to the internal assets that need protection.

Armed with these mental models, we began to analyze the systemic root causes for anti-patterns. Many insights emerged from this effort to move us in the direction of recommended practices. Some of the key insights that inspired security in FuSE output product design so far include:

- Stakeholders can describe intolerable losses as security needs.
- Needs-oriented, loss-driven, capability-based analysis defines security strategies that yield functional requirements that enable effective security tactics.
- Functional perseverance requires engines of situational awareness and resilience to deliver and sustain a system that is secure throughout its lifecycle.
- Systems engineering practitioners and systems security practitioners must work together in a collaborative and synergistic manner using a common language to implement effective security design practices.

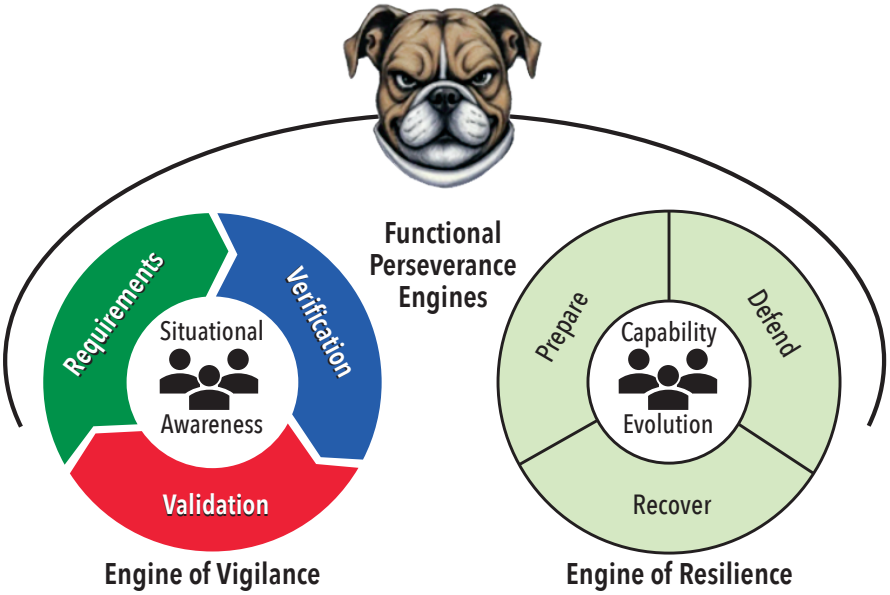


Figure 3. Model for functional perseverance in a hostile predatory environment

The first key insight was understanding that while stakeholders struggle to define security needs, they can describe what aspects of the system they cannot afford to lose. Instead of asking, "what security do you need in the system?" we should ask "what losses can be tolerated and still achieve mission success?".

The second key insight was understanding that security analysis has often resulted in selecting security tactics and techniques without defining a security strategy. When security requirements are described as non-functional requirements, it becomes tempting for the requirements team to select controls to represent these requirements. Without loss-driven analysis, the requirements become a collection of tactics (often implemented as controls) without any regard for a strategy. If the requirements are a collection of controls, then the verification will be a checklist to confirm that the selected controls have been implemented. Loss-driven analysis defines the security needs as loss scenarios. Capability-based analysis defines the security strategies to prevent these losses as functional requirements. Needs-oriented, loss-driven, capability-based analysis starts with security strategies that define system functionality necessary to prevent losses that cannot be tolerated. This is a significant paradigm shift from starting with security tactics to protect internal assets using security controls.

A third key insight was understanding that we need functional perseverance to design a secure system that remains secure in operation. The engine of vigilance is developing the security requirements, verification, and validation to achieve situational awareness. The engine of resilience is the ability of the resulting system to prepare, defend, and recover to achieve capability evolution. The functional perseverance engine shown in Figure 3 is the interaction of these two engines that are perpetually active throughout the system's life cycle.

A fourth key insight was understanding that a collaborative and synergistic relationship between systems engineering practitioners and systems security practitioners is essential. When we highlight desires to increase security proficiency in the systems engineering team, it does not mean the systems engineers need to become systems security experts. It means we want to identify the essential security concepts that comprise an awareness level proficiency in security so that systems engineers embrace a needs-oriented, loss-driven, capability-based approach to defining and validating security needs and defining and verifying functional security requirements. It means developing a common understanding of systems engineering activities that impact security so that systems engineering practitioners and systems security practitioners can work together using common language to implement common practices promoted as part of security in FuSE.

We translated these insights into desired practices for security in FuSE. Some of the practices we have defined so far include:

- Move away from solution-focused tactical requirements (security tactics depicted as security controls) to problem-focused strategic needs (security strategies).
- Use needs-oriented, loss-driven, capability-based analysis to define security as a functional requirement.

When security is expressed as loss-driven needs and functional system capabilities, it is better understood by stakeholders.
- Promote a collaborative effort between systems engineering practitioners and systems security practitioners to design secure systems.
- Functional perseverance results from the interaction between situational awareness (requirements, verification, and validation) and capability evolution (prepare, defend, and recover).

DEMOCRATIZING SYSTEMS SECURITY

As the team explored the intersection of the foundation concepts, the desired outcome evolved to move from solution-focused tactical requirements to problem-focused strategic needs. Stakeholder misalignment became the key concern. We recognized that stakeholder alignment results from a common appreciation of strategic needs that are expressed in terms understandable to all stakeholders. Stakeholder alignment based on an appreciation of needs began to tie all the foundation concepts together.

The IS2023 paper (Dove 2023) recognizes that "systems security is more than a collection of technologists and specialists; it is a mission that needs an aligned team of stakeholders." The paper makes the case for performing needs-oriented, loss-driven, capability-based analysis to define what the system must do in the face of adversity. Defining security needs as protection against the effects of loss in the system provides a meaningful way for stakeholders to articulate what they need the system to do and promotes better stakeholder alignment. The paper suggests that stakeholder alignment can naturally occur around a needs-oriented, loss-drive, capability-based security strategy.

GUIDE TO SECURITY NEEDS AND REQUIREMENTS (GTSNR)

A joint project with the Requirements Working Group RWG to develop a Guide to Security Needs and Requirements (INCOSE 2024) began by identifying anti-patterns related to security needs, requirements, verification, and validation activities and their related mental models. The resulting guide codifies the collaboration of systems engineers and systems security practitioners to perform needs-oriented, loss-driven, capability-based analysis resulting in functional requirements across RWG needs, requirements, verification, and validation (NRVV) activities. It harmonizes the RWG NRVV activities with the systems security activities defined in NIST 800-160 (Ross 2022).

The GtSNR articulates the desired pattern where we use loss-driven analysis to define security needs, use capability-based analysis to define security functional requirement to address these needs, verify security requirements by confirming the capabilities mitigate the loss scenarios, and validate the security needs to demonstrate the system prevents losses identified in the loss scenarios. We define security strategies as part of defining the integrated set of needs. We define security tactics as part of the transformation from needs to design input requirements (system level requirements). We define security techniques as part of the transformation from design input requirements (system level requirements) to design output specifications (system element level requirements).

SECURITY PRIMER FOR SYSTEMS ENGINEERS

The team is working on a Security Primer for systems engineers. The objective of the primer is to motivate, not educate – to show the need for systems engineering attention and show that fulfilling that need is not an onerous task but rather one that fits with accepted systems engineering responsibilities and strategies. The perspective will be functional perseverance in a hostile predatory environment from the perspective of potential system stakeholders. The primer will use the functional perseverance engines shown in figure 3 as the context for describing essential concepts for systems engineers related to systems security.

SYSTEMS SECURITY AWARENESS MICRO-CREDENTIAL

Systems security awareness is one of the micro-credential pilots being conducted by the INCOSE Certification Advisory Group (CAG). Recognizing that existing security credentials focus on network security and other proficiencies at the security tactics level, we are proposing an INCOSE security micro-credential that depicts awareness level proficiency of essential systems security concepts for systems engineers. The following learning objectives were identified for systems engineers demonstrating awareness level proficiency in systems security:
- Describes systems security tasks related to systems engineering activities throughout the system lifecycle and understands why they need to be integrated.
- Describes systems security terms and concepts.
- Describes systems security principles for trustworthy secure design and vigilant system use.
- Describes how to use needs-oriented, loss-driven, capability-based analysis to define functional security requirements.
- Describes what an assurance case is and how it is used to provide evidence of secure and resilient system capability.
- Describes test and evaluation approaches to verify security requirements and validate security needs.
- Describes the importance of system security and system resiliency to prepare for, defend against, and recover from adversity.
- Describes the difference between security strategies, tactics, and techniques.

The focus is to improve communication between systems engineers and systems security practitioners to promote trustworthy secure design as described in NIST 800-160 (Ross 2022). Security in FuSE practices will become essential concepts in training that supports this micro-credential.

SUMMARY

This article serves as a progress report for the ongoing security in FuSE effort, celebrating the team's accomplishments and teasing at more results to come. It is not too late to join the team. The composition of the team has changed since the effort started as new members expressed an interest in joining and previous members acknowledged time pressures that limited their ability to attend the bi-weekly meetings. Potential team members need to be systems thinkers that can focus on the current state to move toward a future state, have an interest in achieving the systems engineering vision for 2035 related to making security a foundational perspective in systems design, be committed to attending a significant quantity of the bi-weekly meetings, and be actively engaged in the discussions and resulting work products.

This article also serves as a case study in the application of systems thinking concepts to analyze the current practice of security in systems engineering holistically to strategically transform the practice for the future of systems engineering. The security in FuSE team has devoted hundreds of hours in intense discussion and collaborative development to comprehensively understand the mental models driving the current systems security practices and then design an innovative transformation of these practices to achieve FuSE objectives related to security. The progress reported here captures the output products completed and in progress that will become meaningful enablers to make effective systems security in FuSE a reality. ■

REFERENCES

> continued on page 22

Protecting Mission Critical Systems The Need for a Shift in Culture, Strategy, and Process

Ron Ross**, ron@ronrossecure.com; **Kymie Tan,** kymie.tan@jpl.nasa.gov

** Former Fellow, National Institute of Standards and Technology

■ **ABSTRACT**
In contrast to the traditional compliance-based approach to protecting space systems using the NIST Risk Management Framework (RMF), a trustworthy secure systems engineering approach as described in the NIST Special Publication 800-160 is proposed as a viable and effective alternative. This paper discusses the issues and concerns with the traditional approach to cybersecurity and how engineering-based approaches measurably improve security, allowing a greater return on investment for mission critical operational environments like those that support space missions. The paper will show that there are several facets to the cybersecurity problem that go beyond the technical to include culture, process, and policy, and explain why a change in strategy and approach is necessary to address the modern sophisticated cyber adversary operating in a world of highly complex and evolving systems. Insights from a project where a NIST SP 800-160-based engineering approach was applied to secure a space mission will be discussed. The early lessons not only illuminate the benefits of security systems engineering, but also the effect of culture, policy and process on building resilience into mission critical systems.

■ **KEYWORDS:** trustworthy secure systems; secure-by-design; systems security engineering; cyber-resilient systems; securing space systems; assurance; systems engineering; security design principles; advanced persistent threat; authorization-to-operate; mission risk; system life cycle

INTRODUCTION

Space is an essential component to the modern economy and vital to the national and economic security interests of the United States. The space sector is critical to many industries, including telecommunications, navigation, and the defense industrial base. Engineering trustworthy, secure space systems is a significant undertaking that requires a substantial investment in the requirements, architecture, and design of systems, components, applications, and networks. A trustworthy secure space system is engineered to provide compelling evidence to support claims that it meets its stakeholder requirements to deliver the capability, protection, and performance needed by the organizations investing in the technology. Adopting a disciplined, structured, and standards-based set of systems security engineering activities and tasks provides an important starting point and forcing function to initiate a needed change toward defensible space systems that are resilient to the modern adversary.

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

BACKGROUND

In 2002, the United States Congress passed the Federal Information Security Management Act (FISMA) (Anon. 2014), affirming the government's commitment to protecting the confidentiality, integrity, and availability of federal information and information systems. As part of the FISMA legislation, the National Institute of Standards and Technology (NIST), a bureau

within the Department of Commerce, was given important responsibilities for developing and implementing cybersecurity standards and guidelines for the federal government and its contractors to ensure compliance with the law. In fulfillment of its FISMA responsibilities, NIST developed the Risk Management Framework (RMF) (Joint Task Force (JTF) 2018) and a series of supporting standards and guidelines to help organizations build, operate, and continuously monitor their information security programs. The publications included standards for security categorization (NIST 2004) and minimum-security requirements (NIST 2006), a comprehensive catalog of security and privacy controls (JTF 2020a), and detailed assessment procedures (Joint Task Force Transformation Initiative 2022) to determine if the controls were implemented correctly, operating as intended, and producing the desired effect with regard to enforcing the organization's security policy.

In accordance with FISMA and the Office of Management and Budget (OMB) policy (OMB 2016), the heads of federal agencies were responsible for managing the information security risks associated with operating their information systems. The NIST RMF was the primary vehicle used by agencies to protect the information being processed, stored, and transmitted by their systems. Every federal information system was required to receive an authorization to operate (ATO) prior to being deployed into operational environments to carry out federal agency missions and essential functions. The ATOs had to be signed by the heads of the respective federal agencies or their designated representatives. The ATOs conveyed the information security risk accepted by senior leaders after they had implemented all of the required safeguards and countermeasures (i.e., security controls) needed to protect their information and information systems.

THE PROBLEM

The Risk Management Framework and its supporting publications were designed largely for enterprise information technology (IT) systems. These systems, for the most part, were composed of commercial off-the-shelf hardware, software, and firmware components. This has been the primary focal point for the RMF since its inception in 2005. In subsequent years, the framework and controls were applied to operational technology (OT) systems and IoT devices. While the RMF has been effective in the context for which it was designed, it has been less effective when applied to large and complex systems engineering efforts, for example, in DoD weapons systems and the NASA's space systems. This problem has been exacerbated by the convergence of cyber and physical systems and the emergence of artificial intelligence (AI) and robotics technologies. In addition to the above, cybersecurity has largely been implemented as a separate and disconnected process for the past four decades creating several institutional and generations problems. These include:

- Insufficient alignment with the systems engineering life cycle of complex systems, creating a disconnected process
- Insufficient attention to risks involving cyber-physical assets (e.g., application specific intergrated circuits, FPGAs, programmable logic controllers, robotic actuators, sensors)
- Inadequate integration of cybersecurity risks into the established framework for overall project risks (e.g., safety, reliability)
- Inadequate conversion of current threat intelligence into actionable items by systems engineers
- Questionable protection, ambiguous return on investment (e.g., unknown confidence or assurance against a range of specified threats)
- Inadequate visibility into the underlying system design resulting in insufficient trust and assurance in the system capability
- Ineffective for emerging technologies like AI, autonomy, and cloud-based ground stations, insufficient guidance is provided on how to secure these cutting-edge systems effectively or in a timely fashion.

To address these problems, NIST developed a set of systems security engineering (SSE) tools and approaches to help organizations developing systems for their critical missions. The SSE guidance is contained in NIST SP 800-160, Vols. 1 and 2 (Ross, Winstead, and McEvilley 2022; and Ross, et al. 2021). The engineering-based security approach was designed to help organizations address their protection needs for complex systems, manage the risk of uncertainty during the development process, and provide sufficient evidence to authorizing officials to make informed, risk-based decisions on approving systems for operation. However, despite the comprehensive NIST guidance, organizations have been reluctant to adopt the engineering-based security approach to satisfy FISMA and OMB security compliance requirements. The next sections provide additional details on the foundational concepts of engineering-based security and the experiment underway to address the institutional and cultural problems previously described.

SECURITY FUNCTIONALITY AND ASSURANCE

There are two equally important aspects of protecting systems from adversarial and non-adversarial threats: security functionality and security assurance. Security functionality defines the safeguards and countermeasures needed to protect the organization's missions and the systems that support those missions. Security assurance is the grounds for justified confidence that a claim or set of claims about the systems has been or will be achieved (ISO/IEC/IEEE 2019). Justified confidence is derived from objective evidence that reduces uncertainty to an acceptable level and, in doing so, reduces the associated risk. Evidence is produced by engineering verification and validation methods. The evidence must be relevant, accurate, credible, and of sufficient quantity to enable reasoned conclusions and consensus among subject-matter experts that the claims are satisfied. Assurance is a complex and multi-dimensional property of the system that builds over time. Assurance must be planned, established, and maintained throughout the system life cycle (Ross, Winstead, and McEvilley 2022).

The determination of adequate security should be based on the level of confidence in the ability of the system to protect itself against all forms of adversity—that is, conditions that can cause a loss of assets. These conditions include threats, vulnerabilities, hazards, disruptions, and exposures. Adequate security cannot be based solely on individual efforts, such as performing functional testing, demonstrating compliance, or conducting penetration tests. Judgments of adequate security include what the system cannot do, will not do, or cannot be forced to do. These judgments of non-behavior must be grounded in sufficient confidence in the system's ability to correctly deliver its intended function in the presence and absence of adversity and to do so when used in accordance with its design intent. The basis for such judgments derives from well-formed and comprehensive evidence-producing activities that address the requirements, design, properties, capabilities, vulnerabilities, and effectiveness of security functions. These activities include a combination of demonstration, inspection, analysis, testing, and other methods to produce the needed evidence. The evidence acquired from these activities informs reasoning by qualified subject-matter experts who would interpret that evidence to substantiate assurance claims made. Assurance that also considers other emergent properties that the system may possess such as resilience to faults or adversarial incursions.

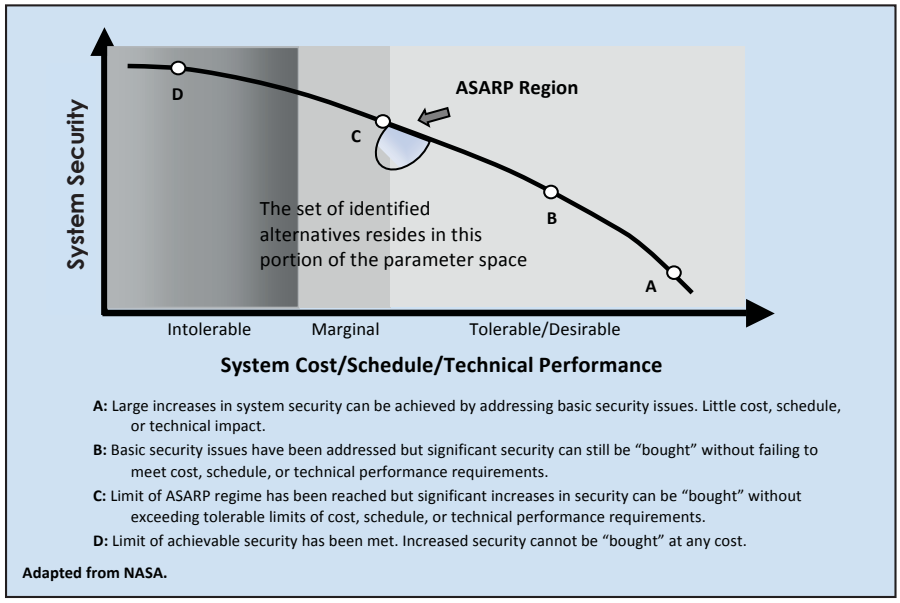


Figure 1. Balancing system cost, schedule, and performance with security

FOUNDATIONAL CONCEPTS

Systems engineering provides a foundation for a disciplined and structured approach to building assured, trustworthy secure systems. Security is an emergent property of an engineered system similar to safety, reliability, and resilience. As a systems engineering subdiscipline, systems security engineering addresses security-relevant considerations intended to produce secure outcomes. The engineering efforts are conducted at the appropriate level of fidelity and rigor needed to achieve trustworthiness and assurance objectives.

In security systems engineering for space systems, mission protection needs guide and inform the selection of security requirements and specifications (i.e., security functionality and assurance requirements). The protection needs focus on: (1) reducing the uncertainty associated with the space system's capability (i.e., system behavior), and (2) controlling (i.e., reducing or limiting) asset loss due to adverse consequences. Adequate security involves a multitude of trade space and risk-based decisions that result in systems that are "as secure as reasonably practicable (ASARP)." Figure 1 illustrates the concept of balancing system cost, schedule, and performance requirements with protection needs.

The foundation of trustworthy, secure systems lies in the security design principles that are applied during the life cycle-based systems engineering process. The principles are described in NIST SP 800-160, Vol. 1 (Ross, Winstead, and McEvilley 2022) and include least privilege, least persistence, least functionality, defense in depth, reduced complexity, anomaly detection, mediated access, domain separation, and least sharing. Security design principles are supported by system and cyber resiliency techniques and approaches as described in NIST SP 800-160, Vol. 2 (Ross, et al. 2021). The techniques and approaches are derived from the security design principles and include, for example, contextual awareness, adaptive response, coordinated protection, analytic monitoring, non-persistence, and monitoring and damage assessment.

NASA/JPL SUNRISE PROJECT OVERVIEW

In the prior section, it was articulated that the foundation of trustworthy, secure systems lies in the application of security design principles during the systems engineering life-cycle process of a system or mission. The expectation is that by doing so, the built system would exhibit improved resilience to faults and adversarial incursions. NIST SP 800-160 describes the security design principles, but not how those principles could be incorporated into well-established, well-exercised, systems engineering processes that underpin many operational systems and projects today.

Several questions arise when considering the application of the design principles such as:

- Where in the system life cycle should key engineering or trade space decisions be made for each security design principle (e.g., it may not be possible to apply certain design principles until critical system components have been built in the later phases of the systems engineering life cycle)?
- What approach or framework can systems engineers use to reason between operational resilience, safety, and security?
- What if the cost for engineering resilience into a mission turns out to be prohibitively high despite producing a quantum of resilience to an attack?

Furthermore, for operational systems like many of those in the U.S. critical infrastructure, the issues of cost, schedule and performance must also be part of the systems engineering decision parameters.

To explore this query toward achieving the desirable outcome of a more secure, resilient system, NASA/JPL in collaboration with NIST undertook a pilot experiment aimed at studying how the design principles for building trustworthy secure systems in NIST SP 800-160 could be incorporated into a well-established systems engineering process for space flight missions. The fundamental questions of interest for the experiment included:

- Can the security design principles in NIST SP 800-160 be integrated into the systems engineering life cycle of an operational system to produce a trustworthy secure system?
- How much improvement can be expected with respect to security when compared to the current approach that uses the NIST RMF and baseline security controls selected from NIST SP 800-53 (JTF 2020a) and NIST SP 800-53B (JTF 2020b)?

The mission selected for the NASA/JPL pilot was SunRISE, a composition of six CubeSats that work together to study solar activity. The science objective of the mission is to better understand how the Sun generates solar particle storms that can be hazardous to spacecraft and astronauts.

In undertaking the SunRISE systems security engineering pilot, a few fundamental challenges were identified in advance. Among the more prominent—the challenge of decomposing the design principles in NIST SP 800-160 into executable engineering actions that will integrate into the well-established, systems engineering life cycle of the SunRISE space flight project. Although the principles in NIST SP 800-160 have been established for some time, the constructs, models, processes, and frameworks needed to translate the principles into concrete engineering activities are largely absent in the literature, industry standards, and/or widely-accepted best practices.

Another notable challenge was that the SunRISE satellite project had to account for the pragmatic considerations of an engineered system deployed in a real-world context—namely cost, schedule and performance. In applying the NIST SP 800-160 design principles to an operational system, real-world constraints also had to be

considered in concert with the security and resilience of a built system. An operational system will necessarily include mission critical requirements, mission objectives, safety and reliability constraints, and other key considerations. All of these elements are necessary to achieve "mission resilience," an emergent property of an engineered system similar to security. Consequently, the SunRISE pilot experiment not only measured the security properties of the engineered system, but also other mission-essential considerations such as cost, schedule and performance.

The next sections will describe the approach taken in the design of the SunRISE experiment where the intention is to illuminate and support the central premise of this paper: that a shift toward the addition of sound security systems engineering is needed to produce trustworthy secure systems that can more effectively address today's adversaries. Additional technical details regarding the design of the SunRISE experiment including the significant number of engineering decisions and parameters that were employed, will be provided in future publications.

EXPERIMENTAL APPROACH

The following sections describe the experimental approach for the NASA/JPL SunRISE pilot project. These include the hypotheses, objectives, scoping criteria, and high-level methodology for the experiment.

Experiment Hypotheses

The overarching hypotheses for the SunRISE pilot established the basis for the experiment.

Hypothesis 1: Systems Resilience

- A systems engineering approach based on the application of the security design principles in NIST SP 800-160 produces a system that is more resilient and secure than a system that uses the traditional NIST RMF and pre-selected baseline security controls.

Hypothesis 2: Support for Risk-Based Decisions (Authorizations to Operate)

- A systems engineering approach based on the application of the security design principles in NIST SP 800-160 provides the necessary and sufficient assurance evidence to support credible risk-based decision making and the requirements for a system authorization to operate (ATO).

Hypothesis 3: Resources Required

- A systems engineering approach based on the application of the security design principles in NIST SP 800-160 can significantly reduce the level of effort, cost, time, and resources required to achieve an ATO.

Experiment Objectives

The following objectives for SunRISE pilot are intended to test the experiment hypotheses:

- Demonstrate a working use case of applying the security design principles in NIST SP 800-160 to an actual flight project.
- Identify potential protection gaps in traditional cybersecurity approaches versus engineering-based security approaches.
- Identify potential security-related system design and implementation changes.
- Document the cost and effectiveness of engineering-based security.

Experiment Scoping Criteria

The experiment focused on the Ground Data Systems (GDS) component of SunRISE satellite system. The GDS is responsible for collecting and distributing the most valuable asset of the mission: the data. Several factors contributed to the choice of the GDS, the most prominent being that the SunRISE GDS operated in the cloud and could be easily replicated (i.e., creating a digital twin) for the purposes of this experiment.

Experiment Methodology

The high-level methodology for the SunRISE experiment is as follows:

- Identify the system under investigation
  - Identify a NASA/JPL mission that had already achieved its ATO
  - Identify a critical component (subsystem) of the mission
  - **Note**: The critical component selected needs to lie within the resource capacity allocated to the pilot (affordability) – the ground data system (GDS) for SunRISE
- Generate the replica of the system under investigation
  - Produce an exact replica of the SunRISE GDS (digital twin
    Twin A – the original SunRISE GDS
    Twin B – the exact replica of the SunRISE GDS
- Establish the metrics
  - System performance (e.g., CPU resources, memory requirements)
  - Security performance (e.g., mean time to detection, mean time to remediation)
  - Programmatic (e.g., cost, schedule allowances, additional procurements)

1) Establish the baseline
   - Conduct a functional evaluation of Twin A and Twin B to ensure that the GDS functionality, resource usage, and system behaviors are identical between both instances (no attacks)
2) Select applicable security design principles from NIST SP 800-160
   - Principles selected based on SunRISE GDS architecture, mission requirements, and NIST guidance
   - Included Least Privilege, Least Sharing, Least Functionality, Mediated Access, Least Persistence, Anomaly Detection, Reduced Complexity, Defense in Depth
   - Also included resiliency techniques and approaches mapped to the security design principles
3) Design attacks against Twin B where the security design principles have been applied
   - The attacks were selected based on common security concerns for the GDS such as data exfiltration or the malicious modification of critical data
4) Design and implement defenses for Twin B
   - The security design principles from NIST SP 800-160 were used to design and implement the defenses into Twin B
5) Verification of GDS functionality
   - A functional evaluation was conducted on Twin B to verify that core the GDS functionality remained intact after the applying the NIST SP 800-160 security design principles
6) Execute attacks
   - Both Twin A and Twin B were subjected to the set of designed attacks
7) Collect and analyze results
   - Measurements for the selected metrics were obtained and the results analyzed.

The choice of a NASA/JPL mission that had already obtained its ATO was prompted by the need to compare the difference in security capability between Twin A (evaluated against NIST RMF and the SP 800-53 controls) and Twin B (integrated with defenses guided by the security design principles from NIST SP 800-160). The NIST SP 800-53 control evaluation for Twin A occurred during the mission's Operational Readiness Review (i.e., toward the end of the mission's design and implementation life cycle before launch). This means that Twin B did not "build on" a system already secured by the NIST SP 800-53 control evaluation to show improved security. Rather, the experiment is based on a Twin A and Twin B that were the identical standard NASA/JPL GDS design. The difference being that

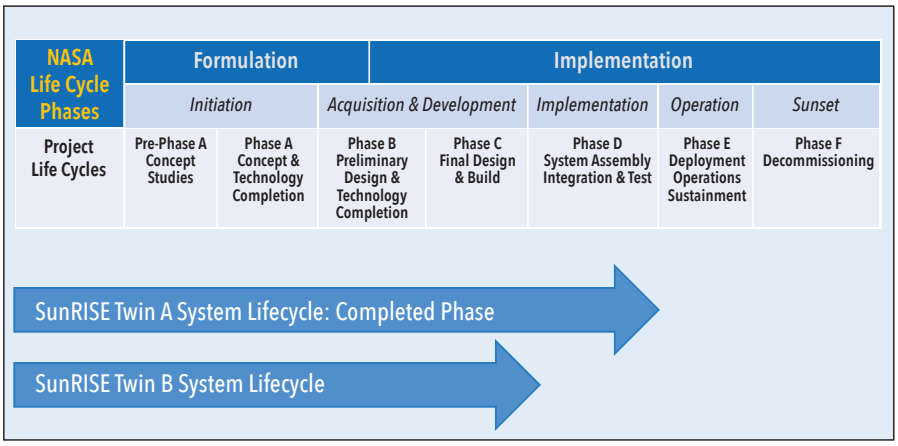


Figure 2. Life cycle phases with Twin A and Twin B

Twin A moved forward to complete the NIST RMF/SP 800-53 controls assessment, while Twin B moved forward to integrate defenses based on the NIST SP 800-160 principles. Figure 2 illustrates the NASA life cycle process with the SunRISE GDS Twin A and Twin B.

Another notable point about the experiment was that the team that designed the defenses using the NIST SP 800-160 security design principles and the team that designed the attacks were separated from each other and did not communicate. This separation helped to ensure that the attacks on the SunRISE GDS were produced independent of the security design principles and implemented defenses.

It was also important that Twin B retained its native function as a GDS despite the modifications introduced by the NIST SP 800-160 security design principles. Consequently, additional tests were executed to continue the comparison of function and resource use between Twin A and Twin B. This was done primarily to ensure that Twin B continued to meet SunRISE mission objectives.

The final step of the process involved a detailed comparative analysis of data collected from both the original (Twin A) and redesigned (Twin B) GDS subsystems, allowing meaningful conclusions to be drawn about the effectiveness of the security capability achieved by the application of the security design principles from NIST SP 800-160.

INITIAL EXPERIMENT RESULTS AND INSIGHTS

The NASA/JPL pilot had only recently completed, and consequently, the discussion in this section describes the preliminary results and insights recorded. These insights tend to revolve around the role of the mission engineers and how they effected the outcome of the project. The initial results also highlighted the difference between the cybersecurity and mission engineering disciplines with respect to the effort to build more trustworthy secure and resilient space systems.

Initial Result #1

A systems engineering approach tightly integrates security functionality into more aspects of a mission, better clarifying the impact and contribution of security to mission objectives.

In the SunRISE system, the traditional security control assessment for the ATO occurred after the mission systems had been designed, implemented and tested. Given the nature of the current security control assessment, this may make sense because at the earlier phases of the system life cycle, there are no implemented systems to evaluate. Implementation of the IT/cyber substrate for the SunRISE GDS is typically conducted in Phase C of a life cycle that begins in Phase A (design) and ends in Phase E (operations).

However, having the cybersecurity assessment occur in the later phases of the system life cycle facilitated a separation, or "siloing" of cybersecurity from the main SunRISE mission. The result was that the cybersecurity engineers who were engaged in the traditional RMF process did not have a strong understanding of the SunRISE mission, its objectives, or the engineering trades and decisions that contributed to the already built system. This lack of understanding resulted in one of the most noteworthy complaints from the mission engineers—that is, the cybersecurity engineers could not articulate how adversarial actions posed a risk to mission objectives, system capability, or how the security controls selected and implemented constituted a measurable reduction in risk to mission success.

Furthermore, because the traditional RMF ATO process was applied in the later phases of the system life cycle, the mission engineers did not have a strong understanding of the how the implemented security controls and artifacts contributed to the mission objectives and system capabilities critical to mission success. The controls and artifacts were perceived as incidental to the already built system.

One of the advantages of the NIST SP 800-160 engineering approach observed during the pilot project is that it engaged the mission engineers early in the system life cycle, specifically at the design phase. It catalyzed engineering questions and considerations with respect to the security-based mission failure implications associated with a specific NIST SP 800-160 security design principle being addressed. For example, consider the design and placement of sensors within a mission system—specifically, sensors that enable the engineers to detect anomalous application behavior with the objective of detecting mission failure. Instead of increasing system complexity and risk by adding cyber-specific intrusion detection sensors into the system, the mission engineers modified the sensors already in use for mission purposes, updated the concomitant operational processes for those sensors, and redesigned the sensor placement to make them dual purpose (i.e. detect potential faults and/or potential adversarial activity). This engineering activity occurred during the design phase and was prompted by the security design principle of "Reduced Complexity." The application of this design principle ensured that security considerations were tightly integrated into the foundational design of the system and resulted in two significant outcomes:

- Because mission engineers were engaged in the modification and placement of the sensors for both reliability and security purposes, they understood the role of the sensors—that is, what part of the system and mission environment these sensors were monitoring and what the output of the sensors would mean to specific mission objectives. This provided the mission engineers with a stronger grasp on how to diagnose a problem or anomaly.
- Because the sensors were developed within the scope of the mission's systems engineering process, sensor operation, maintenance and contribution to the mission workflow was tightly integrated. The mission engineers knew what the output of the sensors meant, they knew how to process that output, report the findings and perhaps more importantly they understood the impact of sensor failure to the mission.

The outcomes described above could

not be attributed to the traditional RMF approach simply because the approach as executed today, does not address how to integrate security considerations into the early phases of the system life cycle (e.g., the design phase).

In addition, the two outcomes above also support the following insights:

- A trustworthy secure systems engineering approach works well because the process involves tightly integrating security functionality into other mission-critical areas and not only on the cyber-related infrastructure.
- A trustworthy secure systems engineering approach is critical to enable rapid detection of adversarial behavior and diagnosis of potential adverse impacts and consequence to mission objectives and capability.

The first point notes that the integration of security into mission systems does not begin and end with implementing security controls only into the system's technical assets. The integration of security functionality must also include elements such as the mission's operational workflows, operational processes like the mission's Anomaly Resolution Process (ARP), and human resources who are able to understand the functionality of the security capability within the context of the mission, cost, schedule, performance, and maintenance considerations. All these elements are naturally addressed in the systems engineering process that properly establishes the security capability in the mission system. This is a fundamental reason why in the sensor scenario above, Twin B (using the NIST SP 800-160 security design principles in a life cycle systems engineering process) was observed to be more effective at addressing an adversarial incursion than Twin A (using the traditional RMF approach).

The second point notes that because the systems engineering life cycle engaged the mission engineers with security considerations from the start, the security functionality was incorporated into the workflow and processes of both mission engineers and operators. This means that when an anomaly or incursion occurred in the SunRISE experiment, the necessary steps to identify, diagnose, and remediate the issue were already "built in" as nominal mission processes, and could be executed rapidly and effectively by the mission team.

Evidentiary support of the two insights is suggested in the preliminary results from the SunRISE pilot:

- The mean time to detect the data tampering attack injected into both Twin A and B was reduced from weeks in Twin A to minutes in Twin B.
- The detection of mission data destruction was reduced from weeks in Twin A to seconds in Twin B.
- A malware-based malicious data tampering incursion executed on both twins was not detected with the traditional RMF approach on Twin A, but was detected with the trustworthy secure systems engineering approach on Twin B.

Initial Result #2

The traditional RMF approach can impede mission resilience and/or success.

The RMF approach applied toward the latter phases of the system life cycle lacks alignment with a mission's objectives and its systems engineering life cycle, creating a disconnected process. This disconnect meant that the mission engineers did not fully understand the security components introduced to satisfy compliance requirements or how to incorporate that security functionality into the space mission's operational profile. This introduced risk to mission resilience and success.

An example of when the RMF approach became an impediment to mission resilience and mission success was observed on the SunRISE pilot. When an anomaly appeared in Twin A and Twin B (an anomaly that was caused by a specific attack introduced to both twins), the mission engineers associated with Twin A saw the anomaly but were confused about what it represented and how they were to address it. Because the security components were introduced into the system late in its life cycle, the mission engineers did not understand the output of those components (such as a SIEM) or the semantics of that output with respect to mission failure/success. In short, the mission engineers did not understand what the presence of security components did to mitigate the threats to the mission or how to interpret the output from those components. It was unclear to the mission engineers where to look, how to understand the security audit data, and what it meant to mission objectives and capability.

Mission operations typically require the rigorous treatment of unknown events and anomalies using a well-established ARP. In the ARP, mission engineers are compelled to address the causal mechanisms underlying a given anomalous event. It took significant effort for the mission engineers to diagnose the anomaly introduced into the SunRISE GDS by trying to understand what the outputs from the security component introduced into the system. This significantly impinged on the natural mission processes that had to take place and consequently risked mission success.

It was observed that although the application of discrete technical controls associated with the RMF served the cybersecurity compliance requirements, it was less clear that they contributed to the overall success of the mission. Furthermore, the SunRISE experiment revealed that where cybersecurity engineers saw the adversarial threat as the primary motivating function for protecting the mission, the mission engineers saw the adversarial threat as merely one of several significant threats that could impinge upon mission success. The other threats would include structural failures, man-made disasters, human errors and so forth. In short, *cyber resilience did not equate to mission and system resilience*. This discrepancy in underlying concepts served to effectively block the successful integration of security into mission systems.

Initial Result #3

Small, inexpensive modifications engineered into mission systems can result in significant gains in resilience against a cyber adversary.

Two examples discussed in this initial result illustrate the consequences of incorporating security early into the system life cycle to address mission objectives. The first focuses on the security design principle of "Mediated Access" that involved the application of anti-virus scans designed by mission engineers. Although the traditional RMF approach did check that anti-virus scans were executed, it couldn't exercise the necessary depth of knowledge to check that the scans were executed on key system components critical to the mission. The mission engineers had that deeper knowledge of not only where in the system to apply the scan but also when in the lifecycle would a scan pose the least risk to mission objectives. Consequently, when they were engaged, they proposed a design of placement and workflow for the scans that were not only more effective than the ones assessed by the compliance-based approach, but also more economical in terms of initial deployment and maintenance costs. The solution designed by the mission engineers was more effective and less costly because it was targeted and intentional. It explicitly addressed the mission's critical assets and critical system life cycle phases (e.g., phases where external project partners were scheduled to deposit data into the mission's critical repository, each deposit was scanned before incorporation into the repository). This effective modification was small and inexpensive when engineered during the design and implementation phases, but it would have been an expensive addition after a compliance-based assessment.

A second example concerns application monitoring, a consideration under the "Monitoring and Damage Assessment"

cyber resiliency approach. Engineering the necessary logging capability to capture application telemetry for identifying adversarial incursions can constitute about 2 to 3 lines of code during the design and implementation phases (e.g., to capture CPU resource usage, memory usage patterns, application communication profiles, etc.). However, if this logging capability were to be added after a compliance-based assessment of a system that had already been designed, integrated, and tested, the cost would be prohibitive, and the issue would be delegated to the list of risk-based decisions that the project must make.

Lessons Learned

The most prominent observation from the NASA/JPL pilot project is that the integration of security into mission systems does not begin and end with the system's technical components. For operational viability, the integration of security functionality must include those areas that are naturally addressed within a systems engineering approach such as the mission's operational workflows, operational processes like the ARP, trained operators who understand the performance and functionality of the security components within the context of the mission itself and the associated cost, schedule, and performance objectives.

CONCLUSION

The traditional RMF approach to cybersecurity and the associated ATO process works extremely well on enterprise IT systems that use mostly commercial off-the-shelf products. However, for certain types of systems being developed for high-intensity, mission critical operations such as NASA space flight systems, DoD weapons systems, and other high-value assets in the U.S. critical infrastructure, a systems engineering approach is needed to help ensure that security is treated as an emerging property of a mission system and integrated into the system life cycle. NASA/JPL conducted an experiment on the SunRISE satellite space flight system to determine if applying the security design principles from NIST SP 800-160 as part of a disciplined and structured system life cycle process, could result in more effective protection for the space system. After executing the traditional cybersecurity RMF process and completing the control assessments necessary to achieve an ATO, a comparison was made to the same system (i.e., a digital twin) that used a carefully selected set of security design principles from NIST SP 800-160. The initial results were extremely promising with respect to the engineered system that embodied the design principles. By applying the security design principles early in the system life cycle as part of an engineering process, the SunRISE mission engineers had increased visibility into the system architecture to facilitate better placement of the selected security safeguards and allowed those safeguards to be more effective against an adversarial threat. The mission engineers were also able to reduce the complexity of the SunRISE GDS which also contributed toward achieving a trustworthy secure system that was more resilient. The initial results from the experiment prompted NASA to move into the second phase of the experiment, selecting a more complex space flight system and exercising additional security design principles from NIST SP 800-160. The complete SunRISE GDS results will be published and made available at the future publication. ■

ACKNOWLEDGEMENTS:

The content has not been approved or adopted by NASA, JPL, or the California Institute of Technology. Any views and opinions expressed herein do not necessarily state or reflect those of NASA, JPL, or the California Institute of Technology.

REFERENCES

- Anon. 2014. Federal Information Security Modernization Act (P.L. 113-283). s.l.:s.n. Available at: https://www.govinfo.gov/app/details/PLAW-113publ283. [Accessed 3 May 2025]
- International Organization for Standardization/International Electrotechnical Commission/Institute of Electrical and Electronics Engineers. 2019. ISO/IEC/IEEE 15026:2019 Systems and Software Engineering – Systems and Software Assurance – Part 1: Concepts and Vocabulary. Geneva, Switzerland: International Organization for Standardization.
- Joint Task Force Transformation Initiative. 2022. Assessing Security and Privacy Controls in Information Systems and Organizations NIST Special Publication (SP) 800-53A, Rev. 5, Gaithersburg, US-MD: National Institute of Standards and Technology. Available at: https://doi.org/10.6028/NIST.SP.800-53Ar5. [Accessed 3 May 2025]
- Joint Task Force. 2018. Risk Management Framework for Information Systems and Organizations: A System Lifec Cycle Approach for Security and Prviacy. NIST SP 800-37, Rev 2, Gaithersburg, US-MD: National Institute of Standards and Technology. Available at: https://doi.org/10.6028/NIST.SP.800-37r2. [Accessed 3 May 2025]
- Joint Task Force. 2020a. Security and Privacy Controls for Information Systems and Organizations. NIST Special Publication (SP) 800-53, Rev. 5., Gaithersburg, US-MD: National Institute of Standards and Technology. Available at: https://doi.org/10.6028/NIST.SP.800-53r5. [Accessed 3 May 2025]
- Joint Task Force. 2020b. Control Baselines for Systems and Organizations NIST SP 800-53B, Gaithersburg, US-MD: National Institute of Standards and Technology. Available at: https://doi.org/10.6028/NIST.SP.800-53B. [Accessed 3 May 2025]
- National Institute of Standards and Technology. 2004. Standards for Security Categorization of Federal Information and Information Systems. Federal Information Processing Standards Publication (FIPS) 199, Washington, US-DC: U.S. Department of Commerce. Available at: https://doi.org/10.6028/NIST.FIPS.199. [Accessed 3 May 2025]
- National Institute of Standards and Technology. 2006. Minimum Security Requirements for Federal Information and Information Systems.Federal Information Processing Standards Publication (FIPS) 200., Washington, US-DC: U.S. Department of Commerce. Available at: https://doi.org/10.6028/NIST.FIPS.200. [Accessed 3 May 2025]
- Neumann, P. G. 2004. Principled Assuredly Trustworthy Composable Architectures, Menlo Park, US-CA: SRI International. Available at: http://www.csl.sri.com/users/neumann/chats4.pdf. [Accessed 3 May 2025]
- Office of Management and Budget. 2016. Office of Management and Budget Circular A-130, Washington, US-DC: Office of Management and Budget. Available at: https://www.whitehouse.gov/wp-content/uploads/legacy_drupal_files/omb/circulars/A130/a130revised.pdf. [Accessed 3 May 2025]
- Ross, R. et al. 2021. Developing Cyber-Resilient Systems: A Systems Security Engineering Approach NIST SP 800-160 Vol. 2 Rev. 1, Gaithersburg, US-MD: National Institute of Standards and Technology. Available at: https://doi.org/10.6028/NIST.SP.800-160v2r1.
- Ross, R., M. Winstead, and M. McEvilley. 2022. Engineering Trustworthy Secure Systems NIST SP 800-160 Vol. 1 Rev. 1, Gaithersburg, US-MD: National Institute of Standards and Technology. Available at: https://doi.org/10.6028/NIST.SP.800-160v1r1. [Accessed 3 May 2025]

ABOUT THE AUTHORS

Ron Ross**
RONROSSECURE, LLC
Wildwood, FL, USA
ron@ronrossecure.com
** Former Fellow, National Institute of Standards and Technology

Kymie Tan
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA, USA
kymie.tan@jpl.nasa.gov

Systems Security Working Group continued from page 14

REFERENCES

- Dove, R., K. Willet, T. McDermott, H. Dunlap, D. P. MacNamara, and C. Ocker. 2021. "Security in the Future of Systems Engineering (FuSE), a Roadmap of Foundational Concepts." Paper Presented at the 31st Annual INCOSE International Symposium, Virtual, 17-22 July.
- Dove R. 2022. "Setting Current Context for Security in the Future of Systems Engineering." *INSIGHT* 25 (2): 8-10.
- Dove, R., M. Winstead, H. Dunlap, M. Hause, A. Scalco, K. Willett, A. Williams, and B. Wilson. 2023. "Democratizing Systems Security." Paper Presented at the 33rd Annual INCOSE International Symposium, Honolulu, US-HI, 15-20 July.
- INCOSE. 2021. *Systems Engineering Vision 2035: Engineering Solutions for a Better World*.
- INCOSE. 2024. Guide to Security Needs and Requirements. INCOSE-TP-2024-146.
- INCOSE. 2025. *Systems Engineering Competency Framework* (2nd Edition).
- Monat, Jamie P., and Thomas F. Gannon. 2017. *Using Systems Thinking to Solve Real-World Problems*. College Publication.
- NIST Joint Task Force. 2020. NIST SP 80-53 Revision 5 Security and Privacy Controls for Information Systems and Organizations.
- Ross, R., M. McEvilley, and M. Winstead. 2022. NIST SP 800-160 Volume 1 Revision 1 Engineering Trustworthy Secure Systems, NIST.

Guide to Security Needs and Requirements – Making Security a Functional Requirement

Beth Wilson, wilsondrbeth@aol.com

■ **ABSTRACT**
The INCOSE Systems Security Working Group completed a 2-year project to create a Guide to Security Needs and Requirements targeting both the systems engineering practitioner and the systems security practitioner to help them collaboratively define security needs and requirements that result in a secure system in operation. Starting with a set of anti-patterns for security requirements, we identified existing tactics that have not produced secure systems in the operational environment. The team then identified an approach to perform needs-oriented, loss-driven, capability-based analysis across the systems engineering activities. The result is a set of need statements capturing the stakeholder expectations concerning security and a set of functional requirements defining what the system must do to address those needs. Defining security as a functional requirement helps us design a system that can prepare for, defend against, and recover from adversity to achieve and sustain mission success.

INTRODUCTION

The INCOSE Systems Security Working Group (SSWG) and Requirements Working Group (RWG) launched a joint project in October 2022 to explore the application of the RWG technical products to security needs and requirements. As shown in Figure 1, the RWG envisioned the addition of "other domain specific guides" in their portfolio of technical products. The Guide to Security Needs and Requirements (INCOSE GtSNR 2024) was published in August 2024 and represents the first of what is expected to be many such guides.

SSWG has been working on strategic concepts related to security in the Future of Systems Engineering (FuSE) (Dove 2022) defining objectives and strategies to develop and evolve practices. "Security as a functional requirement" is one of these strategic concepts that emphasizes movement away from security as a non-functional requirement (NFR) and is a key focus of the Guide to Security Needs and Requirements. The strategic concept "security proficiency in the systems engineering team" is also a focus where the guide promotes the collaborative interaction between systems engineering practitioners and systems security practitioners to better define security needs and requirements. Collectively, the strategic concepts of capability-based engineering,

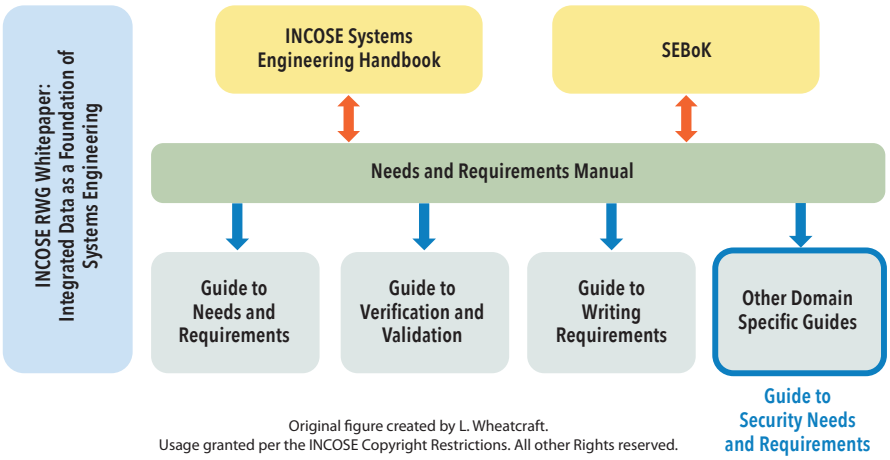


Figure 1. Requirements Working Group (RWG) technical products showing where Guide to Security Needs and Requirements fits as an example of an "other domain specific guide"

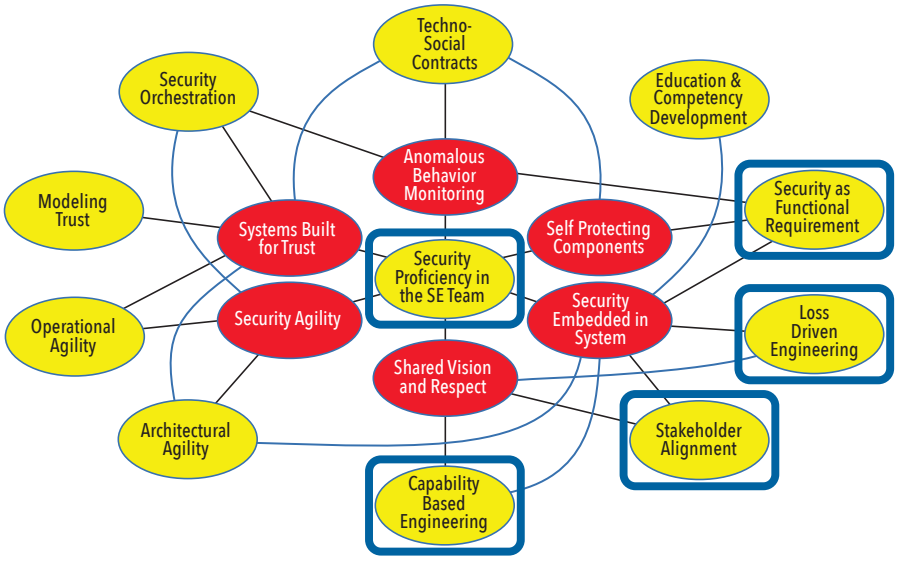


Figure 2. Guide to Security Needs and Requirements captures the focus of multiple strategic concepts related to security in the Future of Systems Engineering (FuSE)

stakeholder alignment, and loss-driven engineering moved the team toward the recommended needs-oriented, loss-driven, capability analysis to define security needs and requirements.

SECURITY REQUIREMENTS ANTI-PATTERNS

During the project kickoff, the joint team reviewed the RWG needs, requirements, verification, and validation (NRVV) concept overview shown in Figure 2 and identified existing approaches in the NRVV context that have not produced secure systems in the operational environment. We defined these as anti-patterns to collectively understand how current approaches fail to produce secure systems in operation. We used these to develop an effective pattern we could describe in the planned Guide to Security Needs and Requirements. We explored the following anti-patterns:

- One security expert
- Security review after design
- Security controls
- Non-functional requirements.

"One security expert" anti-pattern: While we do not want the systems engineering team to punt all the security requirements to the systems security practitioners, the desired engagement needs to recognize that there are many aspects of systems security as a discipline. It is important to understand that systems security includes information assurance, software assurance, hardware assurance, anti-tamper, supply chain risk management, operational security, and others. An information assurance expert with information technology credentials should not be consulted about requirements for anti-tamper capabilities. A software assurance expert should not be asked to allocate requirements to a vendor statement of work through the supply chain for a hardware element. One security subject matter expert cannot cover all the security disciplines. The mental model in play here is, "we had THE security expert look at it, so we're all set." (See Table 1)

"Security review after design" anti-pattern: A systems engineering team may recognize that there are multiple security disciplines that need to be engaged to prevent the previous anti-pattern. They risk creating a new anti-pattern if the engagement with systems security practitioners is after the design is complete. The time to engage the systems security practitioners is during needs analysis, requirements development, and architecture definition, not after the design is complete. If the systems security practitioners identify security gaps and vulnerabilities after the requirements have been defined, it can be a daunting effort to change them to address these shortfalls. For example, if the systems security practitioners identify a need to segment and separate networks after the architecture is complete and all the design output specifications are complete, it may be too late to address an identified vulnerability. The mental model in play here is, "they identified some scary scenarios, but it is too late to change anything." (See Table 2)

"Security controls" anti-pattern: The National Institute of Standards and Technology (NIST) Special Publication (SP) 800-53 "Security and Privacy Controls for Information Systems and Organizations" (NIST 2020) provides a catalog of security tactics and techniques organized into control families. While it is a helpful taxonomy of approaches, it is often misused as a checklist for compliance.

It is helpful to have consistent language. For example, the "role-based access control" noted as AC-3(7) is part of the access control family (AC) in the access enforcement control enhancement (AC-3) and is among other types of access enforcement controls. There are important differences between access control (AC-3), separation

Table 1

Security anti-pattern	One security expert
Needs	Want a secure system
Requirements	NFRs created/reviewed by one security subject matter expert (SME)
Operational system	Not secure
Mental model	We had THE security expert look at it, so we're all set.

Table 2

Security anti-pattern	Security review after design
Needs	Want a secure system
Requirements	Shall be secure
Collaboration	Systems security practitioner on review team
Operational system	Not secure
Mental model	The security experts identified some scary scenarios, but it is too late to change anything.

Table 3

Security anti-pattern	Security controls
Needs	Want a secure system
Requirements	NIST SP 800-53 controls
Verification	Checklist of controls
Operational system	Not secure
Mental model	If we want to get the security certification, we need to make sure we include ALL the controls.

Table 4

Security anti-pattern	Non-functional requirements
Needs	Want a secure system
Requirements	Shall do XYZ securely
Design	Basic security measures
Verification	Confirm basic security measures are in place
Operational system	Not secure
Mental model	Just add a firewall and require passwords.

Table 5

Security pattern	Needs-oriented, loss-driven, capability based functional requirements
Needs	Loss-driven analysis
Requirements	Treat security as a functional requirement
Design	Identify security gaps in mission threads
Verification	Confirm capabilities to mitigate loss scenarios
Operational system	Mission success
Mental model	Systems engineers work with system security practitioners to perform needs-oriented, loss-driven, capability-based analysis to define security functional requirements.

of duties (AC-5), and least privilege (AC-6). The security controls help systems engineering and systems security practitioners use a precise and consistent terminology when describing these techniques.

When security requirements are described as non-functional requirements, the resulting absence of real security needs makes it tempting for the requirements team to select controls from the NIST SP 800-53 catalog instead. There is no strategy to address the security need, only tactics to map controls to the requirements. When the requirements become a collection of security controls, then the verification becomes a list of those controls inviting a checklist compliance approach.

To make matters worse, there are some teams that decide more is better and they open the controls closet and pull everything out. Implementing role-based access control in one area of the system so that AC-3(7) can be checked off does not consider the other places where it should also be applied. Finding a way to include all the controls in the catalog "just to be safe" is expensive and does not make the system more secure. With this anti-pattern, the resulting system is not adequately secure, protected, or resilient when in the operational environment. The mental model in play here is, "if we want to get the security certification, we need to make sure we include ALL the controls." (See Table 3)

"Non-functional requirements" anti-pattern: When security is defined as a quality characteristic, it is tempting to view security requirements as non-functional requirements. If there is a requirement to store data, then the NFR is to store data securely. What does this really mean? How can we implement it? How do we verify it? The ambiguity makes it tempting to define basic security measures such as firewalls, encryption, password protections, etc. as part of the design to satisfy the NFR. The resulting system is not secure in the operational environment. Treating security requirements as NFR means that we deliver a system that can achieve the functional use cases but leaves the system vulnerable to loss scenarios that were not considered. The mental model in play here is that if we add a firewall and require passwords, we can satisfy the NFR to store data securely. (See Table 4)

SECURITY NEEDS AND REQUIREMENTS DESIRED PATTERN

After exploring the security anti-patterns, the team was able to develop the desired pattern with the NRVV context. The desired approach is a needs-oriented, loss-driven, capability-based analysis where we define a set of security functional requirements.

Focusing on the needs, we perform loss-driven analysis to define what losses can be tolerated and still achieve mission success. Which mission threads are critical? How can these survive an attack by an adversary? Needs analysis is loss-driven analysis.

Focusing on the requirements, we treat security as a functional requirement. We ask what the system must do to address the needs we identified. We develop loss scenarios and misuse cases. We then identify system capabilities to prevent these losses. Requirements analysis is identifying the capabilities to mitigate the loss scenarios we established during needs analysis.

During verification, we confirm that these capabilities mitigate loss scenarios. During validation we confirm that the operational system can achieve mission success.

With this pattern, we have achieved a mental model that focuses on needs-oriented, loss-driven, capability-based analysis. We now have a mental model where systems engineers are responsible for system security and that security is a functional requirement. (See Table 5)

The NIST SP 800-160 "Engineering Trustworthy Secure Systems" (Ross 2022) describes the concept of loss and the need to define security capabilities as systems security concepts. The standard includes appendices related to lifecycle processes and technical processes that

describe security activities and tasks that can align with the NRVV concepts. The desired pattern is that systems engineering practitioners collaborate with the systems security practitioners while performing these security activities throughout the systems engineering lifecycle.

STRATEGY VS. TACTICS

Emerging hardware and software security design approaches provide innovative ways to address system element assets that need protection. With complex systems (and systems of systems), it becomes important to also consider the system-level security strategies to ensure that these security designs will be effective in the operational environment. If the system is not architected to reduce attack vectors in its system elements, then emergent behavior at the system level introduces vulnerabilities that cannot be mitigated with even the most comprehensive security tactics at the hardware and software level. As systems become larger and more complex, it becomes cost-prohibitive and ineffective to harden system elements after they are designed and built. At the system level, it is no longer enough to identify tactics (security controls) to satisfy non-functional requirements where security is a quality characteristic.

The real issue with the security controls and non-functional requirements anti-patterns is that security controls are a collection of techniques to implement security tactics. While effective at the system element level, at the system level these anti-patterns are implementing tactics

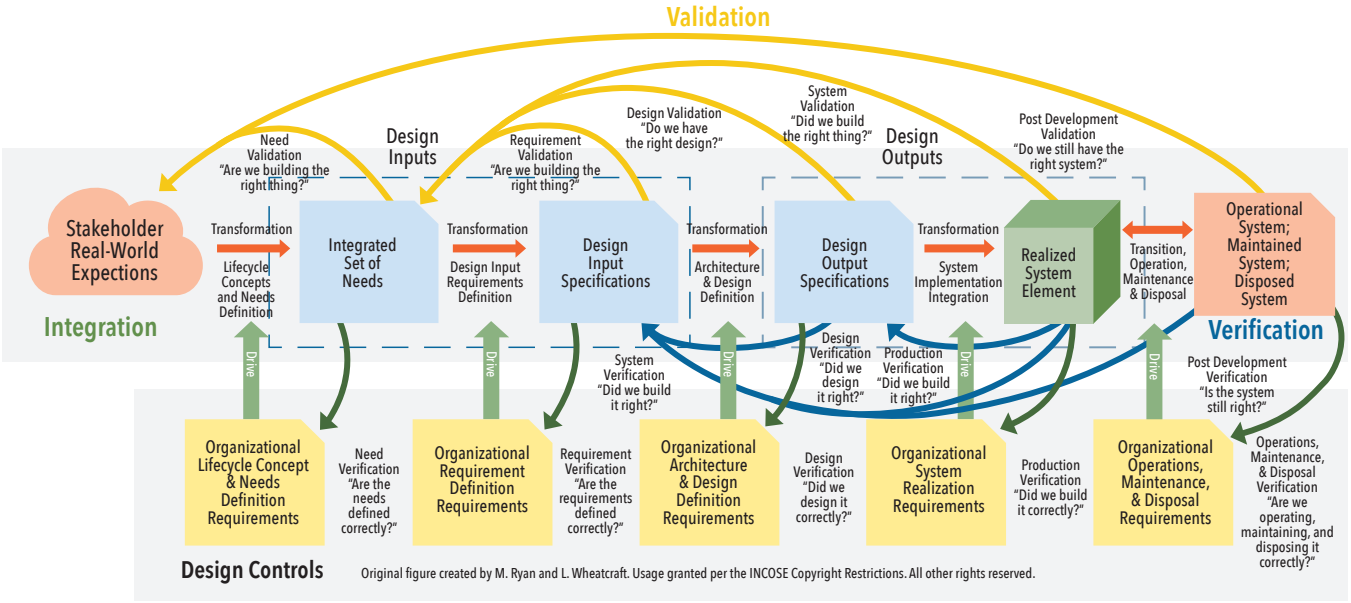


Figure 3. Needs, requirements, verification, and validation (NRVV) concept

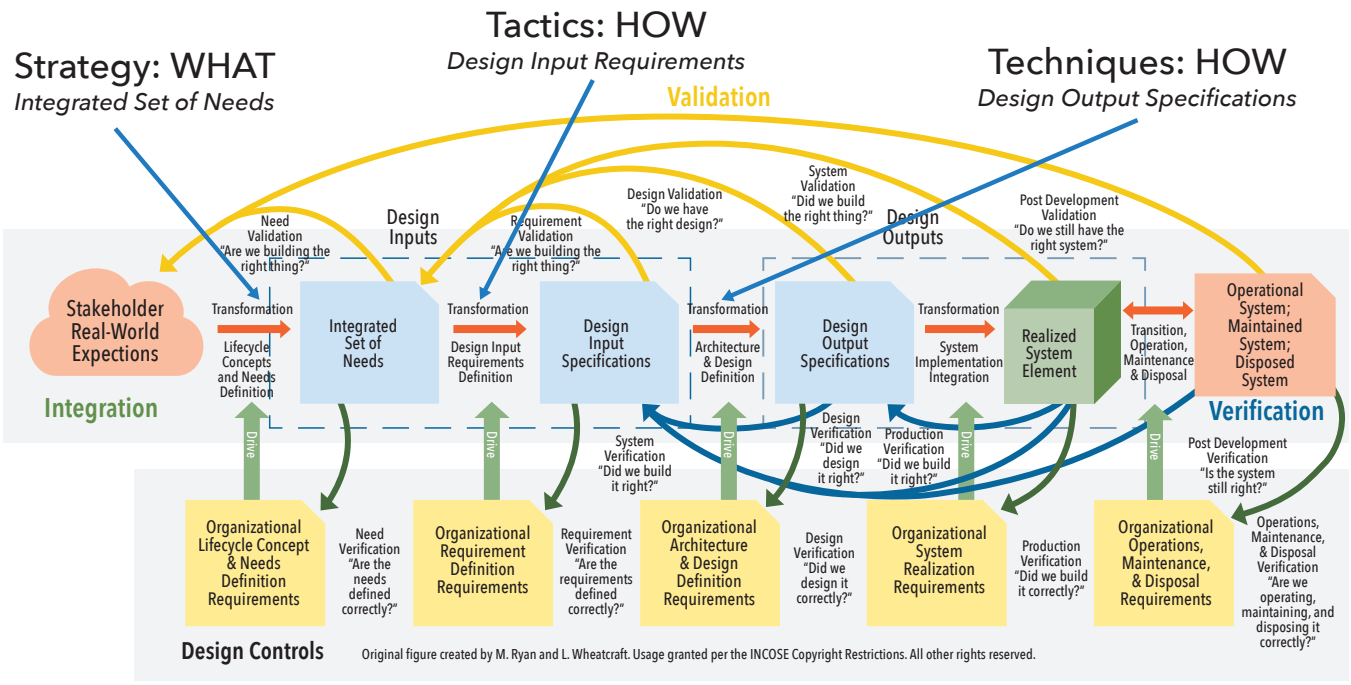


Figure 4. How security strategies, tactics, and techniques map to the needs, requirements, verification, and validation (NRVV) concept

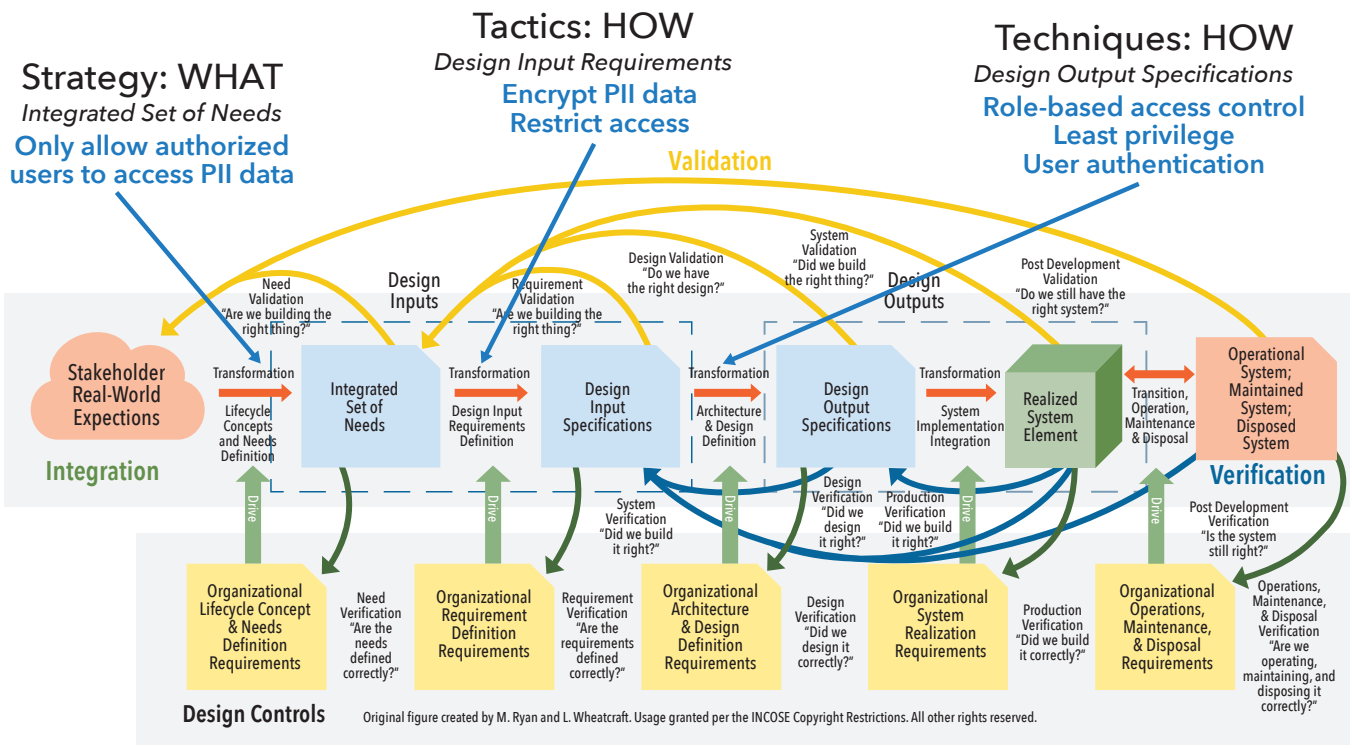


Figure 5. Example of a security strategy identified while defining the integrated set of needs. Two tactics are identified while defining design input requirements to achieve this strategy. Three techniques are identified while defining design output specifications to achieve one of these tactics.

without a strategy. We find the following quote from Sun Tzu's book *The Art of War* (Tzu 2010) relevant for this: "Strategy without tactics is the slowest route to victory. Tactics without strategy is noise before defeat." (See Figure 3)

Figure 4 shows that we need to define security strategies while defining the integrated set of needs. We define security tactics to achieve these strategies while defining the design input requirements. We define security techniques to achieve these tactics while defining the design output specifications.

Security strategies result from the needs analysis as part of the transformation from the stakeholder real-world expectations (captured as loss scenarios) into an integrated set of needs. These security strategies define what the essential functions are that must be secured against disruption. These define what losses can be tolerated. We represent the resulting security strategies as need statements. This is the "what" part of the secure design.

Security tactics define how to achieve a strategy. Security tactics result from the requirements analysis as part of the transformation from the integrated set of needs to design input requirements. This is the "how" part of the secure design.

Security techniques define how to achieve a tactic. This is another "how" part of the secure design focused on the assets in the system elements. We represent security tactics and techniques as requirement statements.

We can map security strategies, tactics, and techniques to the NRVV concept in Figure 4:

- Strategy: WHAT – Integrated Set of Needs
- Tactics: HOW – Design Input Requirements
- Techniques: HOW – Design Output Specifications.

We can demonstrate this with a simple example shown in Figure 5. One security strategy is to only allow authorized users to access Personally Identifiable Information (PII) data that is considered sensitive data that must be protected. Among the security tactics to achieve this particular security strategy are encrypting PII data and restricting access to PII data. If we follow the example for the restricting access tactic, we can define techniques at the system element level to achieve this tactic including role-based access control, least privilege, and user authentication. We represent security strategies as need statements and security tactics and techniques as requirement statements:

- Strategy: only allow authorized users to access PII data
- Tactics: encrypting PII data, restricting access
- Techniques for restricting access: role-based access control, least privilege, user authentication.

OVERVIEW OF GUIDE TO SECURITY NEEDS AND REQUIREMENTS

The resulting 26-page Guide to Security Needs and Requirements (GtSNR) describes how to develop a set of needs statements that capture stakeholder needs related to security and a set of functional requirements defining what the system must do to address those needs. The guide identifies NRVV activities and then describes the security focus for each of those activities.

Introduction: This section states that the goal of the guide is to answer the following question about the RWG technical products, "How can these products be used for the definition of systems security needs and requirements?" It explains that the guide is intended for systems engineering and systems security practitioners. The systems security practitioner can use this guide to better understand the RWG products and adapt their guidance to security needs and requirements. The systems engineering practitioner working with systems security practitioners can use the guide to better define security needs and requirements.

Security View of Needs, Requirements, Verification and Validation: This section identifies the shortfalls of defining security requirements as non-functional requirements. It describes the needs-oriented, loss-driven, capability-based approach to developing security functional requirements using a simple example. The GtSNR counters the NFR "the system shall store data security" to instead be a simple need statement, "the stakeholders need the system to protect personally identifiable information data." Design input requirements are then defined as functional requirements including, "the system shall encrypt PII data stored in the system" and "the system shall restrict access to PII data to authorized users only."

Lifecycle Concepts and Needs Analysis: This section relates need statements to protection needs as defined in NIST 800-160 (Ross 2022). The GtSNR describes how to define security needs by developing loss scenarios, assurance cases, and misuse cases. The GtSNR defines a need statement template: [The system of interest (SOI) or stakeholders/users] need [protection need] to ensure [objective] in the event of [potential loss scenario]. The following NRVV terms are explored to define the security focus for each:

- Lifecycle concepts and needs analysis
- Need statements
- Integrated set of needs
- Needs (statement) verification
- Needs (statement) validation.

Needs to Requirements Transformation: This section relates this transformation from needs to functional requirements that translate protection needs into system capabilities and performance to be provided by the system. The design input requirements represent the system-level requirements that are inputs into the architecture and design activities to develop requirements for the system elements. The verification and validation activities described in this section are for the requirement statement verification and the requirement statement validation. The GtSNR defines the requirements statement verification as following the guidance in the RWG Guide to Writing Requirements (INCOSE, GtWR 2023). The GtSNR defines requirements validation as ensuring that the security tactics support security strategies and that the planned security techniques will implement these security tactics. The following NRVV terms are explored to define the security focus for each:

- Design input requirements
- Design input requirements definition
- Requirements (statement) verification
- Requirements (statement) validation.

Requirements Analysis: This section relates the transformation from design input to design output requirements including the architecture definition. Passive security functions that do not exhibit behavior are represented in the system architecture as structure constructs. An example is network segmentation. Active security functions that exhibit behavior are represented in the system architecture by functional constructs. An example is user authentication. The following NRVV terms are explored to define the security focus for each:

- Architecture and design definition
- Design output specification.

Design and System Verification and Validation: This section addresses requirements verification and needs validation. The GtSNR describes using system test scenarios that successfully accomplish loss scenarios to show that security tactics satisfy security strategies. System verification plans include loss scenarios and misuse cases. Mission threads are reviewed to close security gaps. System verification can include security testing techniques such as red team assessments and penetration testing. System validation can use adversity emulation techniques to demonstrate that the system prevents losses identified in the loss scenarios. The following NRVV terms are explored to define the security focus for each:

- Design verification
- Design validation
- Production verification
- System verification
- System validation.

Post Development Verification and Validation: This section focuses on the challenge of making sure that the implemented security requirements continue to satisfy the security needs for the system after it is operational. The GtSNR describes using loss-driven, capability-based analysis to intentionally design systems that achieve sustainable security.

SUMMARY

The Guide to Security Needs and Requirements (GtSNR) is a technical product published by INCOSE representing the collective and collaborative effort by the RWG and SSWG. The NRVV concept is applied to security needs and requirements by performing needs-oriented, loss-driven, capability-based analysis across the NRVV concept activities. The emphasis is on defining security as a functional requirement to help us design a system that can prepare for, defend against, and recover from adversity to achieve and sustain mission success. ■

REFERENCES

- Dove, R. 2022. "Setting Current Context for Security in the Future of Systems Engineering." *INSIGHT* 25 (2): 8-10.
- INCOSE. 2023. Guide to Writing Requirements (GtWR). INCOSE-TP-2010-006-04.
- INCOSE. 2024. Guide to Security Needs and Requirements (GtSNR). INCOSE-TP-2024-146.
- NIST Joint Task Force. 2020. NIST SP 80-53 Revision 5 Security and Privacy Controls for Information Systems and Organizations.
- Ross, R., M. McEvilley, and M. Winstead. 2022. NIST SP 800-160 Volume 1 Revision 1 Engineering Trustworthy Secure Systems, NIST.
- Tzu, S. 2010. *The Art of War.* Capstone Publishing.

ABOUT THE AUTHOR

Dr. Beth Wilson is an INCOSE Expert Systems Engineering Professional (ESEP) and INCOSE Systems Security Working Group co-chair. She retired from Raytheon after 33 years as a systems engineer and is currently an adjunct professor at Worcester Polytechnic Institute in their graduate systems engineering program.

Governance and Resilience: A Holistic Approach to Systems Security in Complex and Chaotic Environments

Sue Caskey, sacaske@sandia.gov; and **Adam Williams,** adwilli@sandia.gov

ABSTRACT

A systems governance approach emphasizes a holistic perspective that identifies and navigates the interdependencies and conflicts between security and operational needs. Governance is defined as a collection of metasystems that provide the necessary constraints and processes to support, steer, adapt, transform, and sustain a system (Keating et al. 2022). Utilizing the Cynefin framework, which distinguishes between simple, complicated, complex, and chaotic environments (Snowden and Boone 2007), the article highlights the challenges faced by nuclear power plants in predatory contexts and the importance of integrating security objectives into governance frameworks.

By incorporating security as a fundamental aspect of governance, the article underscores its significance for persistence, adaptation, and transformation in the face of uncertainty. Additionally, it introduces key heuristics of systems security, such as the importance of context, knowledge-based decision-making, and organization-specific sociological factors (Williams and Caskey 2024). Ultimately, this work provides valuable insights into enhancing resilient operations in complex environments by reinforcing the connection between effective governance and security in systems engineering.

INTRODUCTION

Governance plays a pivotal role in ensuring the resilience of complex systems, this includes the system's ability to be secure, particularly in environments characterized by uncertainty, interdependence, and high consequences. Much like Cook's (Cook 2002) concept of safety as being a characteristic of the system rather than characteristic of their components, security is an emergent property of the system. Nuclear power plants exemplify such complex systems, where security and operational performance—and the interaction between them—are paramount. The anticipated expansion of such facilities (via the "new nuclear renaissance" related to advanced and small modular reactors) suggests a higher likelihood of operating within "predatory environments," consisting of multifaceted challenges like physical threats, cyber vulnerabilities, and sociopolitical pressures. Here, governance frameworks can help such systems adapt to and thrive amidst the complexity and chaos of these challenges.

Governance is understood here as a set of constraints, processes, and feedback mechanisms designed to support, steer, adapt, and sustain the system it oversees (Keating et al. 2022; Keating and Katina 2023). For nuclear power plants, this perspective aligns with recent research out of Sandia National Laboratories that explored new approaches for capturing the complexity, dynamism, and interdependencies of current — and anticipated — security performance needs for complex systems (Williams et. al 2023). By also incorporating the Cynefin framework's distinction between simple, complicated, complex, and chaotic domains (Snowden and Boone 2007), a nuanced governance approach to managing the uncertainty and variability inherent in securing complex systems — including nuclear power plant — emerges.

In this context, resilience is a critical dimension of governance, encompassing persistence, adaptability, and transformation (Caskey 2024), as well as relating to complex system security. These elements

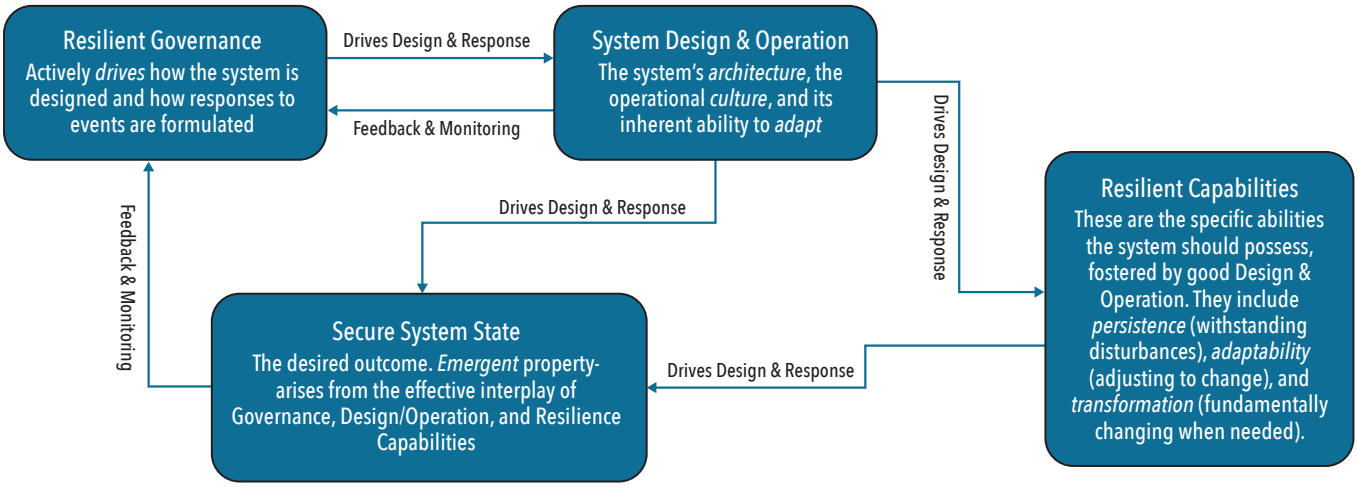


Figure 1. Mental model reflecting relationships between resilience, systems security, and governance

enable systems to maintain functionality under stress, adapt to changing conditions, and evolve to address future challenges. This article leverages theoretical insights from governance frameworks, systems theory, and security heuristics to propose a resilience-based approach for security that is illustrated on nuclear power plants. The approach aligns with emerging paradigms in INCOSE's systems security engineering working group which emphasize trustworthiness, loss-driven strategies, and capabilities-based designs. By advocating for security as an integral part of governance, the paper offers actionable insights for system architects, designers, decision-makers, and operators aiming to enhance system resilience in complex and chaotic environments.

RESILIENCE AND SECURITY IN SYSTEM GOVERNANCE

Traditionally a system's governance was specifically defined to support resilience of the system independent of system security. If, however, security is considered an inherent or emergent property of a system, then we are proposing that a resilient governance directly supports system security, particularly in complex environments. Complex systems engineering defines resilience as the system's capacity to persist, adapt, and transform in response to disruptions; traditional systems engineering reflects that security includes providing protective measures necessary to defend the system from threats (Williams 2020). NIST offers a broader, and more rigorous, treatment of the security and resilience concepts by arguing that each are involved in protecting system capability and functionality (Ross et al. 2022). As such, the range of measures necessary to safeguard against dynamic threats form the foundation for a governance framework capable of addressing the multifaceted challenges inherent in complex system operations.

In the context of governance, resilience is achieved through the integration of persistence, adaptability, and transformation. Persistence involves the system's ability to maintain critical functions under stress, supported by attributes such as redundancy, resource sufficiency, and robust communication channels. Adaptability reflects the system's capacity to adjust to changing conditions, balancing flexibility with stability to ensure continuity of operations. Transformation emphasizes proactive innovation, enabling the system to evolve and address future challenges through learning, transparency, and forward-thinking strategies.

Conversely, traditional approaches to security often focus on such protective measures as physical barriers, cyber defenses, or personnel protocols in isolation. However, an integrated paradigm recognizes security as an emergent property of the entire system (e.g., NIST SP 800-160, Vol. 1, Rev. 1). By leveraging insights from complex system models (like multilayer network models) and governance frameworks, advanced security approaches can identify and address interdependencies across physical, digital, and human domains. Key principles such as situational awareness, graceful extensibility, and trustworthiness underpin such approaches, ensuring that security measures align with the system's broader objectives.

One of the key insights from systems theory is the importance of feedback mechanisms in enhancing both resilience and security (Castelle et al. 2015). Effective governance incorporates feedback loops that enable real-time monitoring, assessment, and adjustment of system operations. These concepts should also be aligned to the security of the system and not only limited to operational performance. For example, environmental scanning processes can detect emerging threats, while communication channels ensure that this information is rapidly disseminated and acted upon. These feedback mechanisms are essential for maintaining situational awareness and enabling timely responses to disruptions.

The Cynefin framework further informs the governance of resilience and security by emphasizing context-specific strategies of systems to changes and uncertainty (Snowden 2017). In simple and complicated Cynefin domains, standardized procedures and expert-driven analysis can address predictable challenges. In complex and chaotic Cynefin domains, responses must prioritize adaptive responses and rapid interventions. This contextual adaptability ensures that governance mechanisms remain effective across a range of scenarios, from routine operations to crisis situations.

By integrating resilience and security into a cohesive governance framework (Figure 1), complex systems can better navigate the uncertainties of their operational environments. This approach not only enhances system performance but also mitigates security risks, ensuring the secure and sustainable operation of these systems.

COMPLEX SYSTEMS GOVERNANCE — A NEW PERSPECTIVE

The complex systems governance (CSG) framework provides a structured approach to managing systems characterized by interdependencies, variability, and multidimensional challenges. Within this framework, governance is not merely a hierarchical mechanism but an adaptive instrument that facilitates persistence, adaptability, and transformation. These three dimensions form the cornerstone of governance resilience, enabling systems to

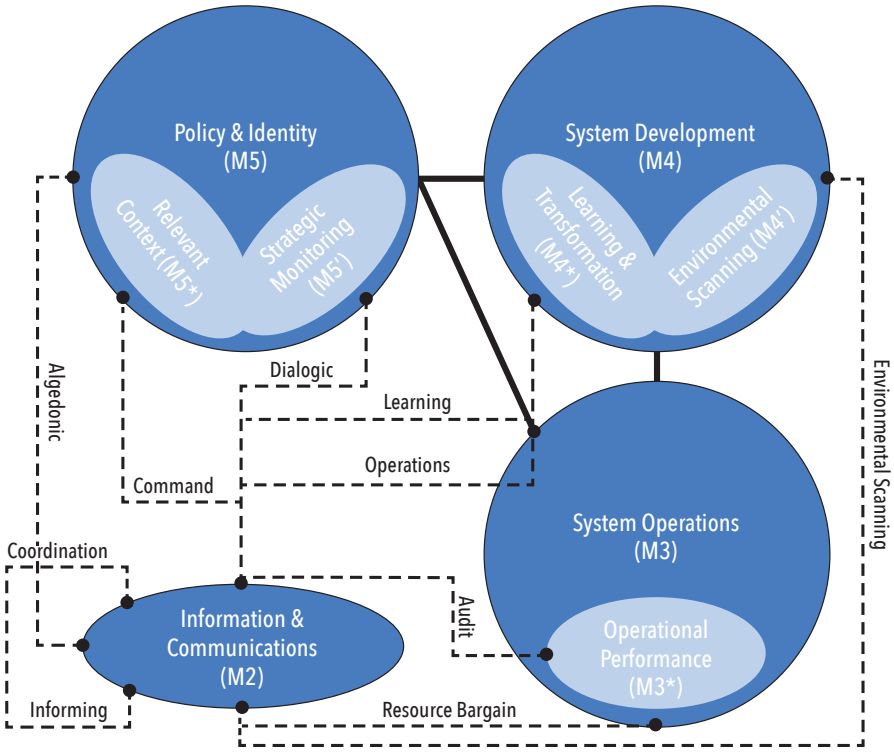


Figure 2. CSG framework (Keating et al. 2022; Keating and Bradley 2015; Keating and Katina 2023)

withstand disruptions, adjust to environmental changes, and proactively evolve to meet emerging demands.

The CSG framework (Figure 2) emphasizes the role of metasystem functions, which provide control, communication, coordination, and integration of a complex system. These metasystem functions ensure that governance not only reacts to immediate challenges but also anticipates and prepares for future perturbations.

Drawing on systems theory concepts such as circular causality, requisite variety, and feedback loops, the CSG framework aligns governance mechanisms with the system's operational and environmental complexities. This holistic approach underscores the need for governance to integrate insights from the system's internal dynamics and external environments.

The Cynefin framework further complements this perspective by categorizing these environments into simple, complicated, complex, and chaotic domains. Effective governance navigates these domains by employing context-appropriate strategies, which can include:

- standardized procedures for simple domains
- expert-driven analysis for complicated domains
- adaptive responses for complex domains
- rapid intervention for chaotic domains.

This contextual adaptability is crucial for ensuring resilience in system operations.

Security is also pivotal in the context of resilient operations. Traditional approaches often treat security as a discrete — and somewhat independent — element of system or facility performance. Recent efforts out of Sandia National Laboratories, however, suggest the potential benefit of a governance perspective that integrates security as an emergent and inherent property of a system (Williams and Caskey 2024). Revisiting Cook's concept of safety being a characteristic of the system rather than characteristic of their components, we postulate that security is an emergent property of the system — and the more complex the system the more dynamic and uncertain the security of the system. Building off empirically derived security heuristics (Williams and Caskey 2024), resilient system operations should inherently support security concerns; specifically, those based on principles such as situational awareness, redundancy, and graceful extensibility. These principles align with the governance resilience emphasis on a holistic perspective and ensures that governance mecha-

Table 1. Cynefin domains and security-governance-resilience mapping

Domain	Security Need	Common Pitfall	Governance Contribution
Simple	Reliability and vigilance against complacency	Over standardization; ignoring signs of change	Reinforce routines, monitor for drift, ensure redundancy, support persistency of the system
Complicated	Expert-driven accuracy and layered defense	Technocratic silos; slow to adapt	Coordinate subsystems, integrate expert feedback, maintain procedural integrity; supports persistency and allows for some adaptability in security
Complex	Emergent threat detection and systemic sensing	Over-control; ignoring emergence; premature certainty	Enable distributed sensing, support adaptive response, foster multi-loop learning; in addition to persistency, governance ensures security adaptability and transformation as needed to maintain a secure state
Chaotic	Immediate containment, fast decision-making	Freezing; overanalyzing; unclear decision paths	Preconfigure authority and response; ensure flexible escalation
Disorder	Orientation and domain identification	Forcing a known frame; ignoring divergent views	Foster plural perspectives, meta-sensemaking, structure for coherence

nisms support both immediate operational needs and long-term system viability.

Synthesizing insights from systems theory, the CSG framework, and the Cynefin model introduces a foundation for a governance-based approach to security. For example, security for systems in the "simple" domain will need to focus on reliability of current solutions and vigilance against complacency within those solutions. Table 1 summarizes a similar treatment for the other Cynefin framework domain. Yet, an over reliance on standardization and susceptibility to change blindness are common shortcomings experienced in these security solutions. In response, incorporating systems governance provides monitoring for drifts away for desired behaviors and support for persistency that help mitigate this pitfall, resulting in overall enhanced security solutions. This foundation provides the basis for exploring practical strategies that enhance system security and adaptability, paving the way for innovative governance approaches for complex systems.

COMPLEX SYSTEM GOVERNANCE USE CASE: ADVANCED NUCLEAR POWER PLANTS

The anticipated future of advanced nuclear power plants presents several challenges that could significantly impact resilience in traditional operations commensurate with the long history of nuclear generated electricity. For example, the wide introduction of so-called passive safety systems (or safety mechanisms that do not require external energy for initiation) may not fully address the dynamic and evolving threats hypothesized for (near) future nuclear operations. Similarly, as advanced nuclear power plants incorporate increased digitization, automation, and remote operations, the operational (and security) landscape becomes more complex. While such advancements can enhance operational efficiency, they can also create new weaknesses in operations that must be identified and mitigated to ensure resilience. These facilities operate in environments where the potential consequences are exceptionally high, requiring governance mechanisms to address vulnerabilities across multiple dimensions effectively.

In response, CSG for advanced nuclear power plants faces a unique set of challenges arising from the intricate interplay of technical, operational, and sociopolitical factors. One significant challenge lies in the technical complexity of future nuclear power plants. Such plants can be conceptualized as systems composed of interconnected physical, digital, and human components, each with distinct vulnerabilities. For example, cyber threats targeting control systems or data integrity can compromise operational safety and regulatory compliance (Williams 2020). Similarly, physical security threats, such as sabotage or unauthorized access, require robust defense mechanisms that integrate seamlessly with primary operational priorities. Governance frameworks must, therefore, coordinate between these diverse functional perspectives to maintain resilient nuclear power plant operations. Specifically, advanced nuclear power plants will consist of novel operational systems, including new nuclear material forms, fuel handling processes, reactor technologies, and ancillary support mechanisms that differ significantly from traditional nuclear reactors. This shift both requires a thorough understanding of the range of operational effects of such changes and the evolution of bespoke resilience (and security) measures adequate to effectively mitigate newly emerging associated risks.

Yet, operational unpredictability further complicates governance. For example, while routine nuclear power plant operations may align with the "simple" or "complicated" domains of the Cynefin framework, unexpected disruptions—ranging from equipment failures to natural disasters—can rapidly shift the environment into the "complex" or "chaotic" domains. The system's governance must adapt dynamically to these shifts, employing strategies that balance immediate response with long-term system stability. Here, the potential for remote, urban, or temporary deployment of advanced nuclear power plants raise additional concerns for operational resilience and security. The complexity and uncertainty introduced by the flexibility of advanced nuclear power plants could result in deployment to locations where personnel may lack extensive experience in nuclear operations and facility resilience.

Another critical challenge involves the sociopolitical context in which nuclear power plants operate. Regulatory frameworks, international oversight, and public perceptions of nuclear energy all exert influence on governance strategies (Bowen et al. 2024). Navigating these external pressures requires a governance system that is not only compliant with stringent regulations but also agile enough to address evolving political and societal expectations. Additionally, the global nature of nuclear oversight necessitates harmonization of governance practices across different jurisdictions, which often have varying priorities and standards. More specifically, national regulatory uncertainty and fledgling international guidance for deploying advanced nuclear power plants may lead to situations where operations, safety standards, and security protocols not sufficiently robust or appropriate.

In addition, these challenges faced by advanced nuclear power plants are substantially impacted by a constantly evolving threat landscape. Consider the previously mentioned anticipated increase in digitization for these advanced nuclear systems. In addition to increased operational efficiency, more digitization and automation also expands cyber and physical attack surfaces, thus creating new vulnerabilities susceptible to potential manipulation. Similarly, wider deployment to needy regions indicates advanced nuclear power plants may be located closer in proximity to a wide array of malicious non-state actors. As the capabilities of such malicious groups improve, the broader deployment of advanced nuclear power plants to remote areas potentially allows more opportunities for sophisticated adversary actions. Lastly, there is noticeable shift in advanced nuclear power plant design related to security, trading the (more costly) tradition of adding "layers" for passive safety to increase security performance. This transformation underscores the benefit of a systems governance approach to incorporate systems security into operational resilience to better mitigate the complexities of modern threats.

By addressing these multifaceted challenges (summarized in Table 2), governance frameworks can enhance the security—and, therefore, the resilient operations—of advanced nuclear power plants. Therefore, the Cynefin domains for advanced nuclear power plants may manifest as operational routines (simple), intricate technical systems (complicated), dynamic interactions (complex), and unexpected crises (chaotic) in remote, urban, or temporary operational environments with no previous experience with nuclear energy. More specifically, a systems governance-based approach leverages insights from systems theory and security heuristics to develop adaptive and holistic strategies for resilient operations in each of these domains.

CONCLUSIONS & IMPLICATIONS

CSG offers a rigorous, logical, and comprehensive approach for incorporating security more intimately into system persistence, adaptation, and transformation. Leveraging core systems theoretic tenets (e.g., feedback processes and circular causality) and insights from current systems models (e.g., the Cynefin framework), governance-based approaches can incorporate security into operational resilience in environments characterized by uncertainty, interdependence, and high consequences.

Table 2. Summary of major challenges and corresponding governance/security responses required for resilient operations of advanced nuclear power plants

Challenge Area	Key Drivers/Features	Implications for Resilience	Governance/Security Response (CSG)
Technical Complexity	Passive safety systems, new materials, reactor designs, digitization, automation	New vulnerabilities; interdependent subsystems; expanded cyber-physical threat surface	Integrated security architecture; resilience-by-design; coordination across physical, digital, and human systems
Operational Unpredictability	Remote or mobile deployment; temporary sites; inexperienced personnel; dynamic environments	Increased risk of domain shifts (from simple to chaotic); limited local response capacity	Adaptive governance frameworks; real-time monitoring; dynamic role and responsibility assignment based on Cynefin domains
Sociopolitical Uncertainty	Regulatory inconsistency; evolving international standards; public perception and acceptance	Potential misalignment between safety/security standards and operational needs	Agile, multi-jurisdictional governance; transparent communication; regulatory harmonization; scenario-based planning
Evolving Threat Landscape	Advanced adversaries; proximity to non-state actors; trade-off of layered security for cost-efficient passive systems	Increased threat sophistication; security assumptions may no longer hold	Systems security as a governance function; feedback-enhanced situational awareness; multi-layered detection and response mechanisms

Here, the anticipated dynamics and trends associated with advanced nuclear power plants exemplify such environments. In response to the inherent focus on responding to disruptions, CSG can help mitigate the complexity introduced by new intrinsic (e.g., new reactor technologies and novel nuclear fuel types) and extrinsic (e.g., remote operating environments and increased digital communications) deployment issues associated with advanced nuclear power plants.

By invoking metasystem functions, CSG provides control, communication, coordination, and integration for resilient system operations across domains, including operational routines (simple), intricate technical systems (complicated), dynamic interactions (complex), and unexpected crises (chaotic). From this perspective, CSG models provide credible pathways for incorporating systems security among difficult cross-dimension interactions between technological complexity, the role(s) of human actors, and non-linear operational environments. Though this article focused on security for nuclear power plants, the underlying logic supports current efforts in the INCOSE systems security engineering community to shift towards an emphasis on ensuring functional persistence of the system in predatory, contested environments. By extension, CSG also affords the opportunity to optimize persistence, adaptation, and transformation efforts to mitigate real-world complexities, dynamic challenges, and disruptive technologies acting against operational system resilience. Advocating for security as an integral part of the system's governance provides insights for enhancing resilient system operations in complex and chaotic environments. ■

REFERENCES

- Bowen, W. Q., M. Cottee, and S. Tzinieris. 2024. "The Evolution of Global Nuclear Security Governance." In *The Oxford Handbook of Nuclear Security*, edited by C. Hobbs, S. Tzinieris, and S. K. Aghara. Oxford University Press. https://doi.org/10.1093/oxfordhb/9780192847935.013.11.
- Caskey, S. 2024. "A Theoretical Framework for Resilience in Complex System Governance (CSG)." Dissertation, Old Dominion University (Norfolk, US-VA).
- Castelle, K., J. Bradley, D. Baugh, and C. W. Chesterman, Jr. 2015. "Systems theory as a foundation for governance of complex systems." *Int. J. Syst. Syst. Eng.* 6 (1/2): 15–32. https://doi.org/10.1504/IJSSE.2015.068805.
- Cook, R. 2002. "How Complex Systems Fail."
- Keating, C. B., and J. M. Bradley. 2015. "Complex system governance reference model." *Int. J. Syst. Syst. Eng.* 6: 33. https://doi.org/10.1504/IJSSE.2015.068811.
- Keating, C. B., and P. F. Katina. 2023. *Complex System Governance: Theory and Practice*. Springer.
- Keating, C. B., P. F. Katina, J. C. Pyne, J. A. Sisti, and A. V. Gheorghe. 2022. "Coupling quantitative vulnerability assessment and complex system governance for systems of systems." *Int. J. Syst. Syst. Eng.* 12: 114–133. https://doi.org/10.1504/ijsse.2022.124979.
- Ross, R., M. Winstead, and M. McEvilley. 2022. Engineering Trustworthy Secure Systems (No. NIST Special Publication (SP) 800-160 Vol. 1 Rev. 1). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-160v1r1.
- Snowden, D. 2017. Liminal Cynefin. Cynefin Co. URL https://thecynefin.co/liminal-cynefin/. (accessed 12.19.24).
- Snowden, D., and M. Boone. 2007. "A Leader's Framework for Decision Making" [WWW Document]. URL https://hbr.org/2007/11/a-leaders-framework-for-decision-making. (accessed 10.9.24).
- Williams, A., and S. Caskey. 2024. "Building a Scientific Foundation for Security: Multilayer Network Model Insights for System Security Engineering." INCOSE Int. Symp. 34: 224–238. https://doi.org/10.1002/iis2.13143.
- Williams, A. D. 2020. "Systems Theory Principles and Complex Systems Engineering Concepts for Protection and Resilience in Critical Infrastructure: Lessons from the Nuclear Sector." *INSIGHT* 23 (2): 14–20. https://doi.org/10.1002/inst.12293.

ABOUT THE AUTHORS

Dr. Sue Caskey is an adjunct professor in system of systems (SoS) engineering at Old Dominion University (ODU). She is a research and systems analyst at Sandia National Laboratories with nearly 30 years of international security expertise. A founding member of Sandia's Global Chemical and Biological Security program, she has supported physical and procedural security assessments and improvements in over 30 countries. Dr. Caskey currently leads analytical projects on global threat prioritization and risk assessment across the chemical, biological, radiological, and nuclear (CBRN) domain—including emerging technologies. Her work supports U.S. Department of State, Department of Energy, and Department of Defense cooperative threat reduction efforts, developing novel tools and models to address current and emerging threats. She also founded a cross-cutting working group at Sandia focused on global analysis and data management. She holds M.E. and Ph.D. degrees in systems engineering from ODU and dual B.S. degrees in biology and computer science from the University of New Mexico. Active in IEEE, INCOSE, and the Society for Risk Analysis, she continues to contribute to the academic and professional systems community.

Dr. Adam Williams is a subject matter expert (SME) on cyber-physical nuclear systems, complex risk for national security issues, and innovative solutions to uncertain global security challenges. Currently, he serves as the Global Security Strategic Studies lead for Sandia's Cooperative Monitoring Center, leads the development of Sandia's Global Security University, provides strategic technical support for various U.S. sponsors (and R&D projects), and is the Nuclear Security & Physical Protection Technical Division chair for the Institute of Nuclear Materials Management (INMM). Dr. Williams also leads and serves as an SME on U.S. Department of Energy's National Nuclear Security Administration (NNSA), Laboratory Directed Research and Development, Electric Power Research Institute, and Department of State (DOS) initiatives. Dr. Williams has a Ph.D. in engineering systems, human-systems engineering from the Massachusetts Institute of Technology (2018). He also has an M.A. in international affairs from the George Bush School of Government & Public Service (2007) and a B.S. in mechanical engineering (magna cum laude, 2004) from Texas A&M University.

A Model-Based Approach for Privacy Risk Mitigation Integrating Systems Engineering with System-Theoretic Process Analysis

David Hetherington, David_hetherington@ieee.org

■ **ABSTRACT**

Certain commercial operations, their systems, and their employees need to operate in hostile or semi-hostile environments. The physical environment may be challenging, but often an unstable political/social environment may be a greater challenge than any temperature or weather extremes. Such an unstable political environment may present rapidly changing threats to employee security. Even if local citizens in the immediate area are supportive, transnational violent gangs may be operating nearby. How do we design overall technology and human systems that can resiliently persevere in such an unstable environment?

Some organizations will reflexively implement a walled-off, fenced, and protected environment for their employees. While this sort of physical protection will be helpful to some extent, if human relationships with the local community are poor or nonexistent, the overall security of the installation will be fragile. Some organizations will deliberately move in the opposite direction, proactively sending their employees out into the community to interact, talk to local citizens, and build human relationships – even when doing so represents a significant degree of physical and personal risk for those employees.

How do we support employees that we are deliberately thrusting into such a risky and unstable environment? For their own safety, we want those employees to communicate as much as possible with the local citizens. We want them to be aware of "chatter" in local social media. On the other hand, we want help them keep their actual personal identity details as protected as possible. Failed social interactions can have lethal consequences. Inadvertently leaked personal data about family members could result in those family members being subject to threats and intimidation in their home location.

In this article, we examine the design of a digital personal communications device designed to achieve these goals and demonstrate the use of System-Theoretic Process Analysis (STPA) in the analysis of a proposed design. Along the way, we will also demonstrate a model-based approach to the design work which represents the recently released standard SAE J3307 "System Theoretic Process Analysis (STPA) Standard for All Industries" (*J3307_202503, 2025*) which specifies an auditable workflow for the STPA methodology originally described in the STPA Handbook.

INTRODUCTION

Many security and safety efforts start from a position of attempting to eliminate all risk. Ever more exhaustive checklists are developed. Attempts are made to exhaustively inspect and harden every component. Employees are restricted to walled compounds. Communication with the local population is restricted or discouraged. Access to local social media communication is restricted or blocked. Such defensive safety postures tend to impair the ability of employees operating in the difficult environment to build trust relationships with the local population. Without such informal communication channels, employees can be blindsided by a violent attack that everyone in the village was aware of days before it occurred.

Of course, regardless of security concerns, employees may need technology support to communicate freely in the local environment. Employees will be unlikely to speak the local language. Even of they speak the local national language, they may not be aware of nuances in the local dialect. Social media communication may likewise

Figure 1. System concept: specialized personal communications device

need substantial technology support to understand local slang, current political context, social mores, and other important communication subtleties.

Employees of global organizations working worldwide are already starting to use mobile phones and common social media applications for translation. While these are a huge step forward from the reliance on paper dictionaries forty years ago, the nuance coverage of such tools is still relatively low. While they might be adequate casual student travel, an organization of might not feel comfortable asking their employees to rely on such devices and applications for their personal safety.

Our future system concept is shown in Figure 1. The key function of the device is to support all forms of local communication the employee might need to develop cordial local relationships and maintain a high degree of local social situational awareness. The personal communication device allows an employee wearing ear bud headphones to engage in multimodal local communication. When talking in person, the device provides on-the-fly, real-time audio translation. The device also supports mobile phone conversations, again providing on-the-fly translation as needed. Finally, the device provides on-the-fly text translation for mobile phone messaging and social media feeds.

On the safety and security side, we also want the device to contain functions to help the employee avoid getting into trouble, either by inadvertently offending the sensibilities of the conversation partner, or by accidentally disclosing sensitive personal identity information. STPA will be very helpful for this part of the design effort.

MODELING THE SYSTEM CONCEPT

While STPA is very powerful for performing a loss-driven engineering analysis, it does not replace the entire systems engineering process. Many considerations for our device such as battery life, thermal performance, shock and vibration, intellectual property rights, and others have nothing to do with STPA. Ideally, STPA should be smoothly integrated with the overall systems engineering process.

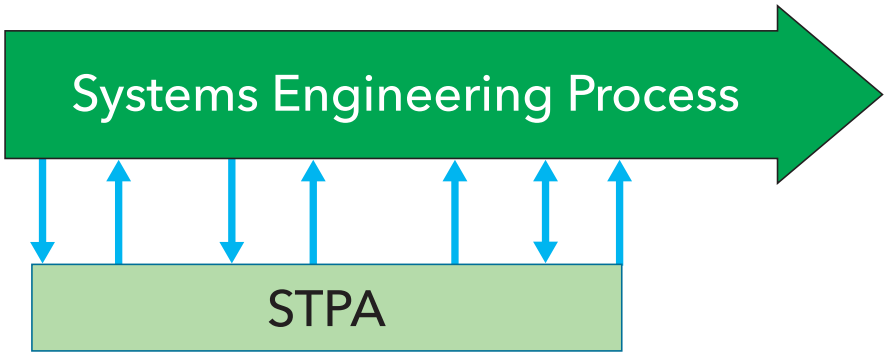


Figure 2. Integration of STPA with the overall systems engineering process

One of the benefits of STPA for security engineering is that it can start much earlier in the life cycle than other common cybersecurity processes. Nevertheless, some amount of basic system definition is required before STPA can begin effectively. For example, before we have identified what sort of system we are building and what context it will be used in, we will have difficulty clearly articulating losses for the system. Likewise, without some general ideas about how the system is expected to operate and interact with its users and environment, it will be difficult to model a control structure or control actions.

On the other hand, systems engineering often starts before the stakeholders have even settled on a clear description of the problem to be solved, much less come to any agreement on a concept of what a system to solve the problem might look like. Supporting the stakeholders on the journey from initial thoughts through to completed "fit-for-purpose" system is a key role of systems engineering for many organizations. Executed properly, model-based systems engineering (MBSE) is merely the process of bringing effective tools to the fundamental systems engineering process to manage complexity and aid in communication. There are a number of different MBSE methodologies providing detailed guidance to systems engineers in effective development of systems models. However, most of these methodologies are variations of the "requirements, functional, logical, physical" (RFLP) method documented by (Baughey 2011) and several other authors (Figure 2).

Looking a little more closely at "RFLP":

- R = "Requirements"– These are not merely "shall" statements in a document, a spreadsheet, or a requirements tool. At this stage, the systems engineer needs to collect all of the "starting line" information about the project. Who are the stakeholders? What are their needs? What program constraints are in place? Which regulations and standards apply?
- F = "Functional"–At this stage, the systems engineer elicits key user stories from the stakeholders describing how the system is expected to interact with its environment. At the "F" stage, we do NOT define a system breakdown structure. The system is seen as a black box. However, we do convert the user stories into function flows. These are usually modeled as (unallocated) activity diagrams. Some organizations prefer to use sequence diagrams for this purpose.

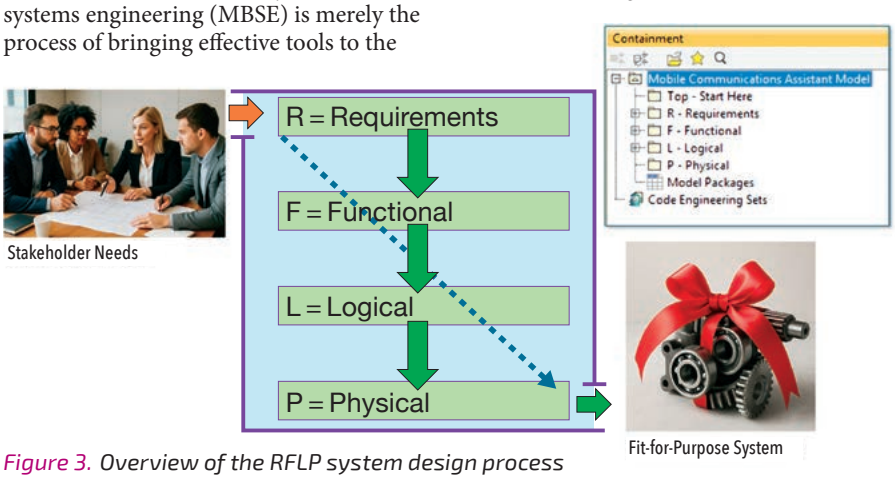


Figure 3. Overview of the RFLP system design process

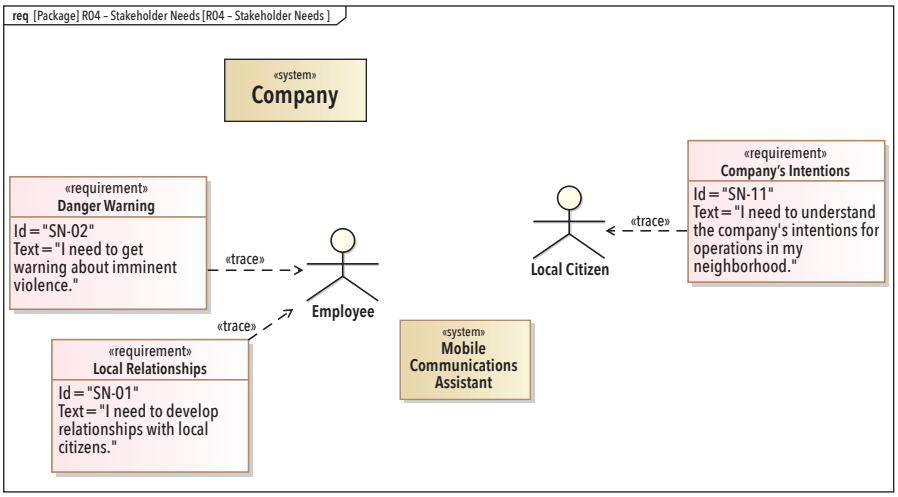


Figure 4. The R layer starts with stakeholder needs

- L = "Logical" – At this stage, we begin to lay out generic "logical" subsystems and components for the system. The next step will usually be a study of possible implementations. At the "L" stage, we keep the names and descriptions of the subsystems and components generic. For example, we might model a "drive-train" subsystem for a car, allowing us to consider both gasoline and battery electric solutions. With the structure of the system laid out, we allocate the functions developed at the "F" stage to the subsystems and components developed at the "L" stage.
- P = "Physical" – We may make multiple physical models of specific real implementations and map them back to the logical system for evaluation.

At our company, we have a generic SysML template for a RFLP project. This template is not linked to any one SysML tool and we make it available to clients unencumbered by intellectual property restrictions on derivative works.

The R layer starts with an analysis of stakeholders and their needs. Figure 4 shows a simplified view of such an analysis. The employee needs to develop relationships with local citizens and receive warning about imminent violence from other actors in the community. The local citizen is suspicious of the long-term intentions of the company in the neighborhood. Is the company going to damage the local environment? Is the company going to upset the social fabric of the area? The local citizen would like to have relationships with company employees to get a more candid feel for what the company has planned. Our stakeholders have decided to invest in a system (The Mobile Communications Assistant) to facilitate the needs of both the company employee and also the local citizen.

As we can see, the systems engineering analysis is not very detailed yet. Nevertheless, it is sufficient for us to start the STPA side of the design (Figure 5). We have identified what we are building and its expected use context. These key pieces of information are enough for us to initiate the STPA process and proceed to J3307 Step 1 in which we begin considering losses, hazards, and system-level constraints.

We will be using MBSE to support both the systems engineering and STPA process.

However, attempting to put every conceivable analysis into a single model leads to complicated models that are difficult to understand and work with. Best practice is to carefully architect a set of linked models that maintain consistency across the effort, but are individually designed to answer specific questions or support specific analysis tasks.

To support this sort of modular modeling approach, our company also maintains a template SysML model structured to match the steps detailed in SAE J3307 (SAE 2025). We also maintain a third SysML model that contains a profile that extends SysML to add elements that match concepts used by STPA such as "control action" or "process model".

The STPA process begins by working with the top-level stakeholders to identify the top-level losses for the system (Figure 6).

Following the style first presented in the STPA Handbook (Leveson 2018), almost all elements in the STPA process are numbered. For example, "L-x" is used as a format for numbering losses, "H-x" for hazards and so on. One of key goals of STPA is to provide requirements to the overall engineering design process. Many organizations use specialized requirements management tools such as IBM's DOORS products to manage these requirements.

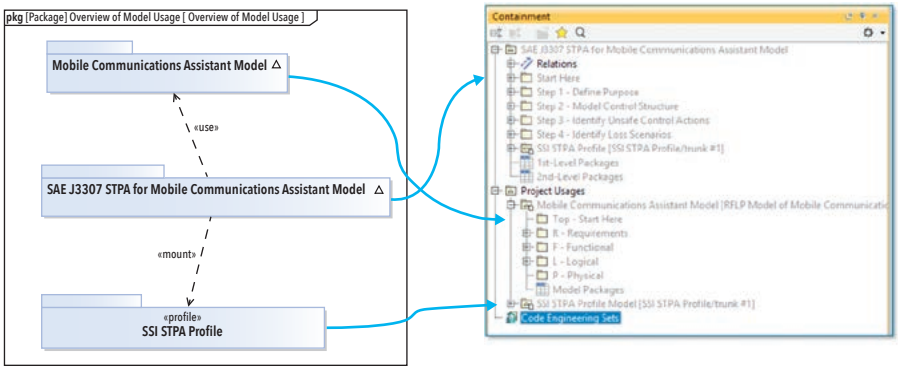


Figure 5. Setting up for STPA

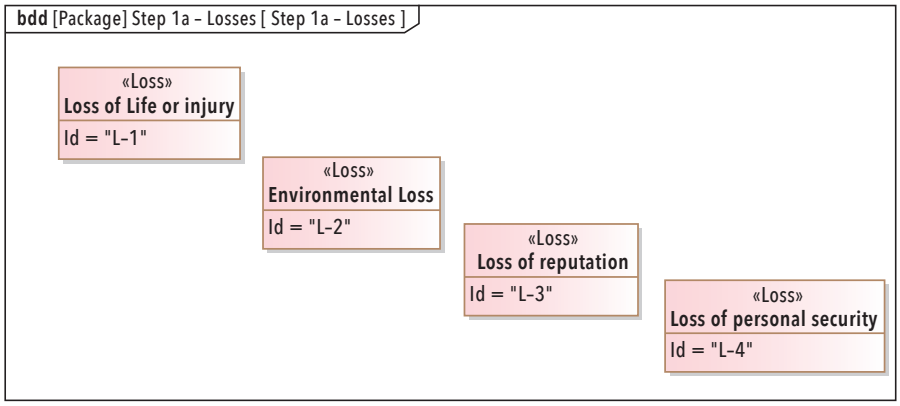


Figure 6. Stakeholder losses

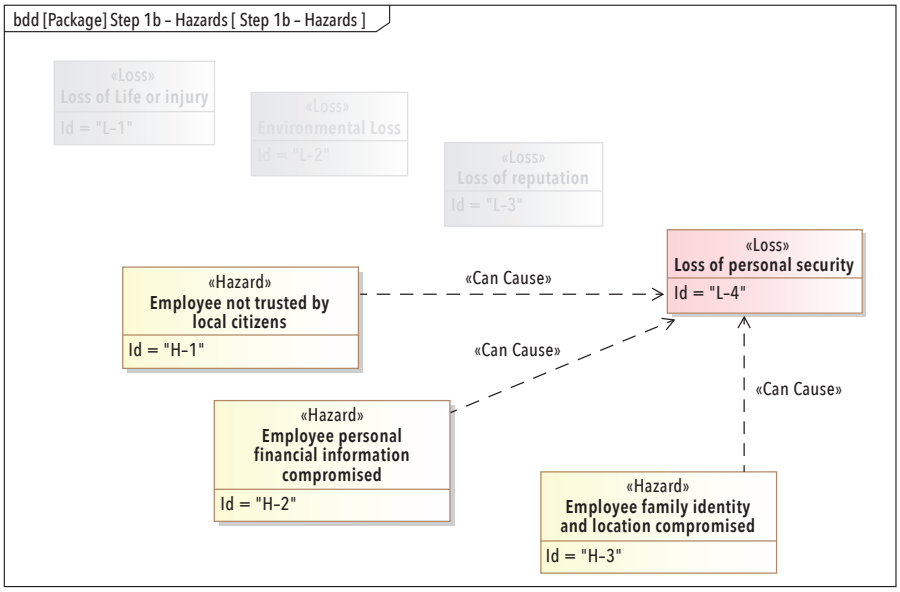


Figure 7. Hazards for one loss

SysML has a specific "requirement" element designed to integrate smoothly with such a requirements management environment. Most SysML tools also support specific integrations to exchange SysML requirement elements smoothly with such large requirement management systems. As such, our STPA profile creates most of the relevant STPA elements such as "loss" and "hazard" by specializing the SysML "requirement" element. This approach should allow most organizations to smoothly transfer outputs from the STPA process into their formal requirements management systems and processes. Readers who are familiar with SysML and the SysML requirement element will spot this pattern without difficulty in Figure 6.

Although often broadly similar, each system and each set of stakeholders will have nuanced concerns. For example, "Loss of life or injury" is usually at or near the top of the list for almost any system. However, in this case, the stakeholders have chosen to focus on the fact that the company is deliberately placing employees in a risky situation. The stakeholders have made "Loss of personnel security" a top-level loss.

Taking a closer look at that loss, let's identify some of the hazards that could cause the loss (Figure 7).

- If the employee loses the trust of the local citizens, they might not warn him about imminent violence.
- If the employee's bank account identity or other personal financial information is disclosed, this information might be passed to criminal networks operating in the area.
- If the identity and physical location of the employee's family members are disclosed, these family members could become subject to threats, intimidation, or worse by transnational affiliates of criminal networks operating in the area.

Having identified the hazards, we can formulate some system-level constraints. These system-level constraints are actually requirements for the system designers.

Here we have formulated a few system-level constraints needed to prevent one of the key hazards, the loss of local citizen trust (Figure 8). These are rather challenging requirements. Our systems engineers are going to have to identify the most powerful artificial intelligence capability that can fit into such a mobile device in order to meet these system-level constraints.

While some of our team has been working on the STPA Step 1 activities, other engineers have been continuing to elaborate the RFLP system model. They have been working with stakeholders to identify use cases and have identified a use case called: "Conversation with Local Citizen".

Examining Figure 9, we can identify one of the small challenges in integrating STPA with the rest of the systems engineering modeling effort. By convention, in systems engineering models actors are external to the system. There are many good reasons for this convention. On the other hand, in

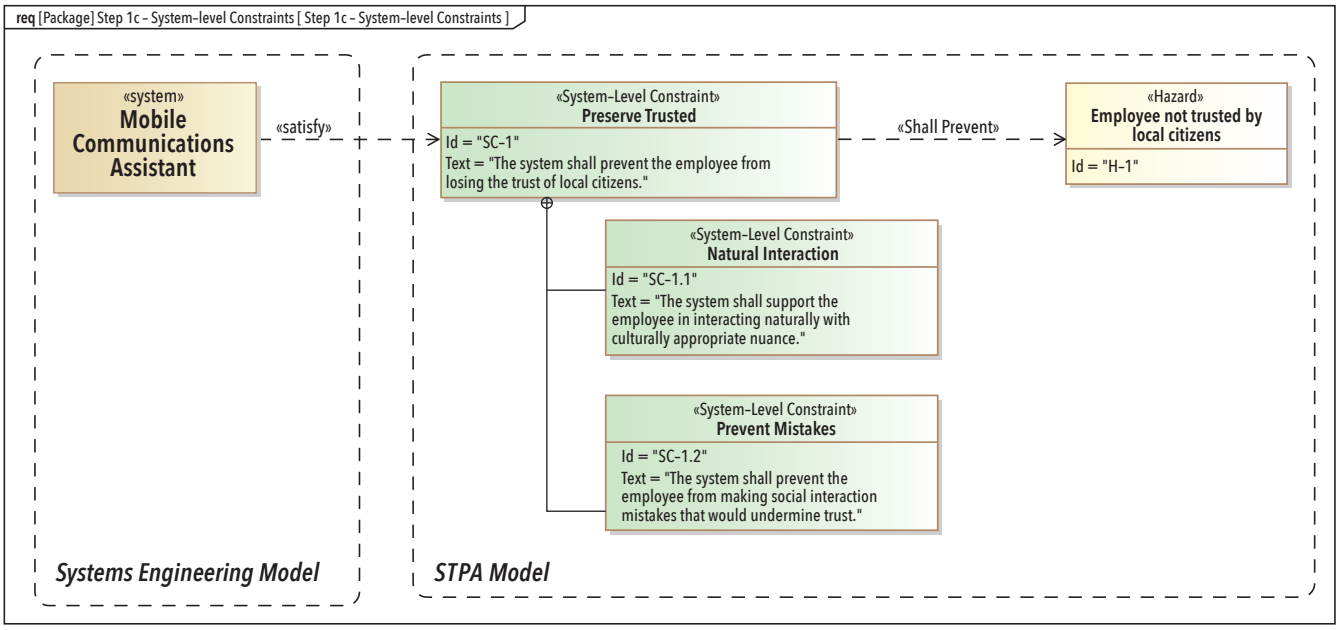


Figure 8. System-level constraints

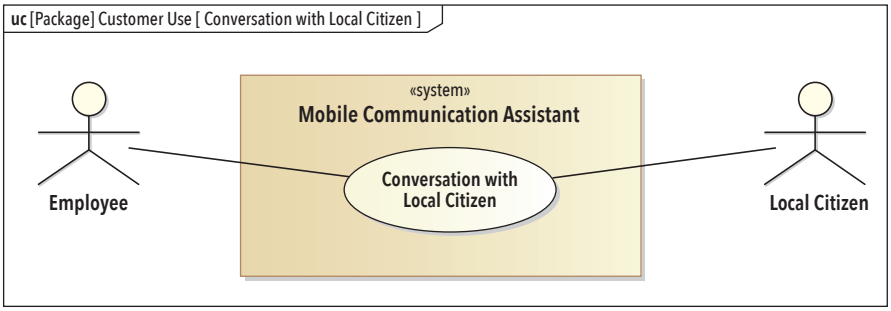


Figure 9. Use case: conversation with a local citizen

the STPA process, actors will be controllers and will be inside the system. This sort of alignment difference need not prevent successful collaboration between a STPA team and a systems engineering team. However, it is worth noting that STPA modeling is not as simple as adding some tags to an existing systems engineering model.

In working with the stakeholders, the systems engineers have elicited the following user story for this use case:

> *"While conversing with the local citizen, the employee talks and listens using special secure ear buds that are connected to the mobile communications assistant using a secure link. As the employee speaks, the arrival of voice sound over the link causes the mobile communications assistant to begin translating the employee's words and social intent. The device translates not only the specific words, but also the employee's nuance and intended social impact. The device generates appropriate voice audio in the local citizen's language. The translated voice audio includes not only the grammatical content from the employee, but also the appropriate tone, intonation, rhythm, and delivery timing. The device accepts the local citizen's voice input and performs the same sort of translation for the employee. Not only is grammatical content translated, but nuance, rhythm, and delivery are matched to the employee's personal cultural expectations."*

The systems engineering team has turned this user story into an allocated flow of actions (Figure 10):

With this level of understanding of the future system, we have enough information return to the STPA side and proceed with STPA Step 2 in which we will define the control structure and control actions for the system.

Working our way through the SAE J3307 STPA process, in Step 2a we define the control structure (Figure 11).

Here we have not yet identified the process models, control actions, or feedback. However, we can already see several interesting aspects of this system. First, there are two controllers for the process. While this is not the main STPA design point, the STPA Handbook does mention commands from other controllers or processes as an effect to consider. The second interesting aspect is that the two humans are both controllers and providing input to the control process. We will want to be careful to separate the voice content from the control action aspects of the conversation.

In a full STPA workup of this system, we would also show the ear buds and the communication device as subsystems of the conversation process. We might even include specific parts of the software as separate subsystems. However, for the purposes of this article the level of detail shown will be sufficient to discover interesting problems.

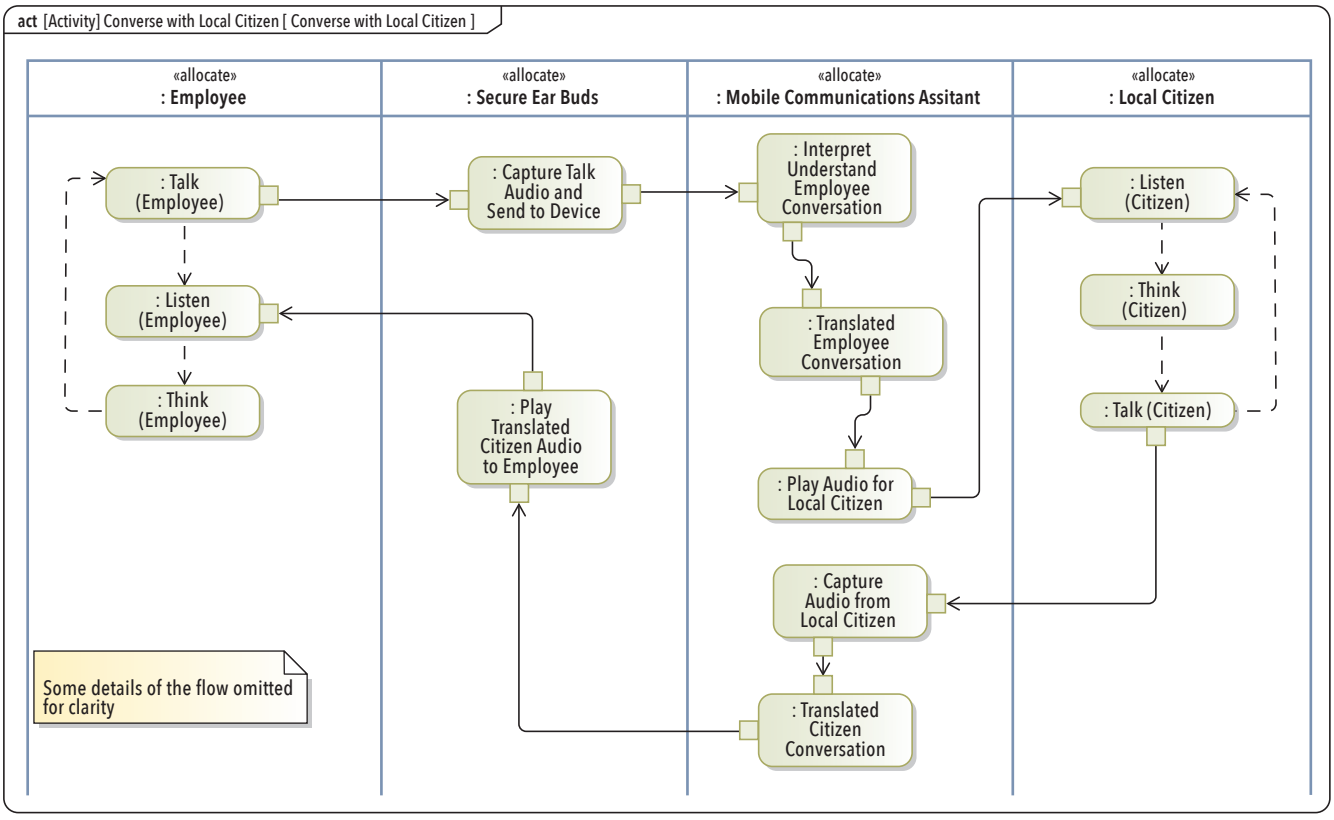


Figure 10. Allocated user story

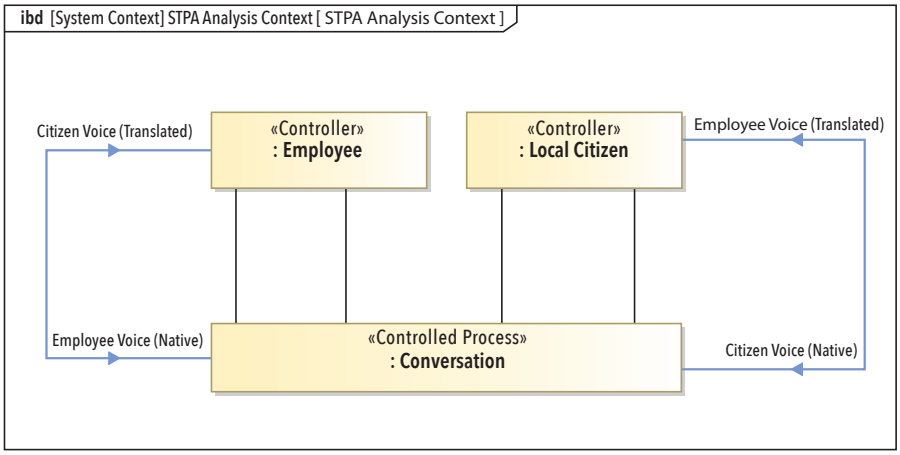


Figure 11. Step 2a: control structure

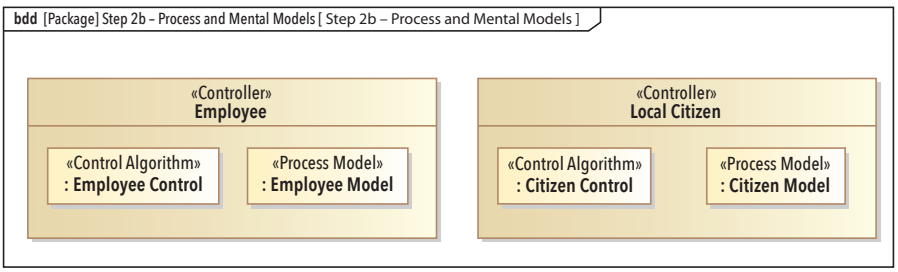


Figure 12. Step 2b: Add control algorithms and process models

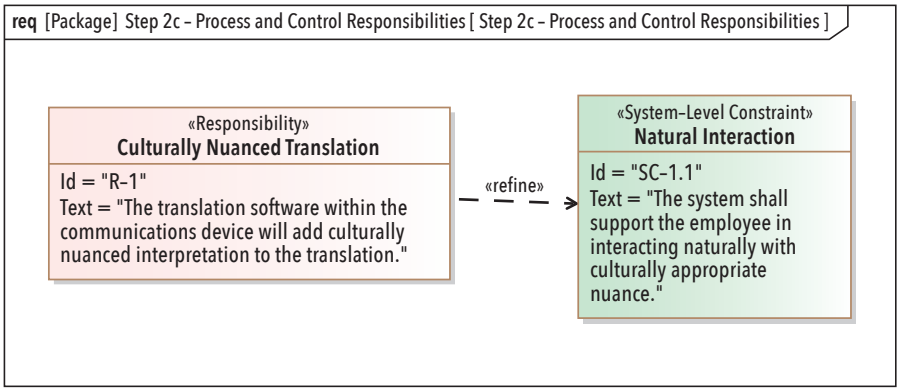


Figure 13. Step 2c: add responsibilities

In SAE J3307 Step 2b we add the control algorithm and process models to our two controllers (Figure 12).

In SAE J3307 Step 2b we add the responsibilities (that is design requirements) for the subsystems (Figure 13).

Finally, we can add control actions in SAE J3307 Step 2d (Figure 14).

Here we have shown only the control actions of the employee or the local citizen speaking. Of the course, in a full STPA workup there would be more control actions. However, for the purposes of demonstrating the STPA methodology, this one kind of control action is sufficient.

If feedback items are apparent from the basic system concept, they can be defined here. Otherwise, the STPA process itself is going to quickly help us identify needed feedback items.

In SAE J3307 Step 3a, we look at four possible standard conditions surrounding a control action to find potentially unsafe control actions. In the case of the employee speaking, it is easy to find examples of all four cases (Figure 15).

In SAE J3307 Step 3b, we can define controller constraints to block the unsafe control actions to the greatest extent possible (Figure 16).

Here we have an example of a controller constraint on the translated software intended to block an unsafe control action. The exercise of defining this controller constraint has identified that we need a feedback item to the employee.

We are now ready for STPA Step 4, loss scenarios. At some level, in completing STPA Step 3, we have finished the problem for the relatively obvious problems. In STPA Step 4 we go looking for the less obvious problems. SAE J3307 follows the process for scenarios described in the STPA Handbook. However, recently John Thomas of MIT has been offering an improved,

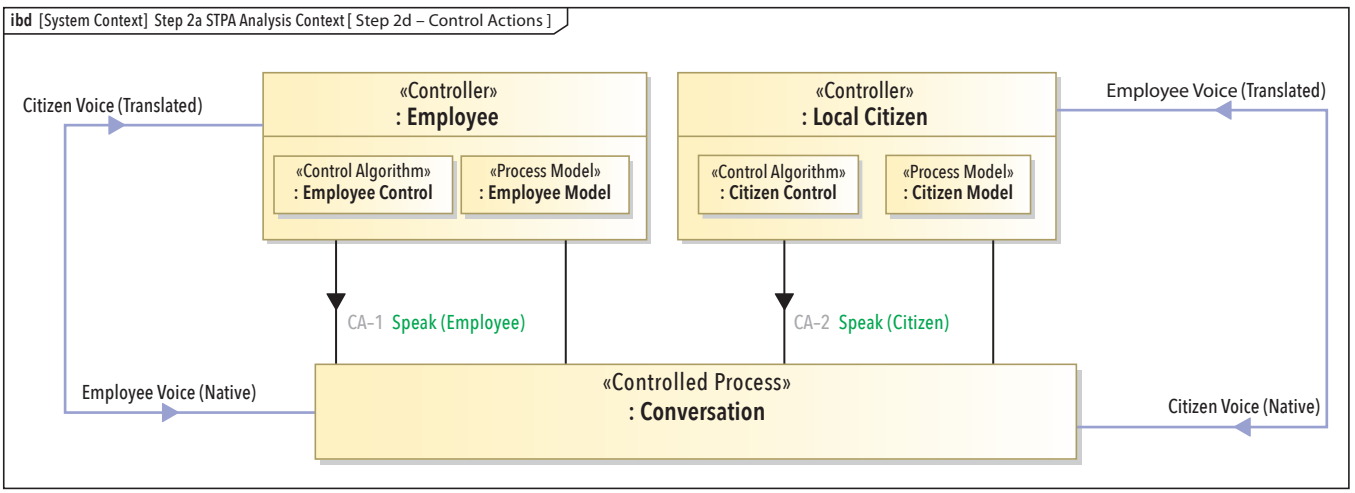


Figure 14. Step 2d: control actions

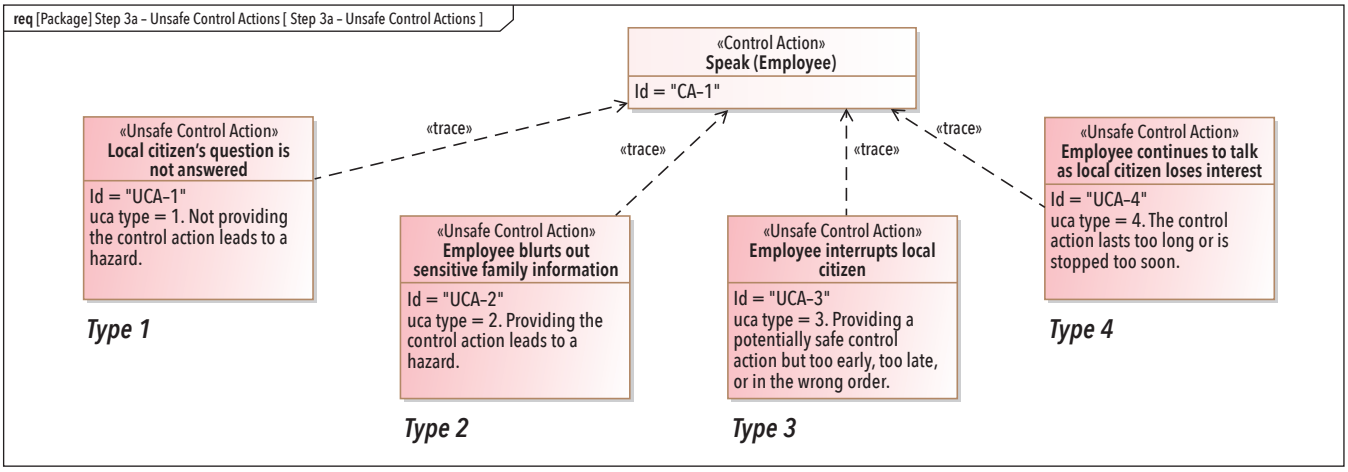


Figure 15. Step 3a: unsafe control actions

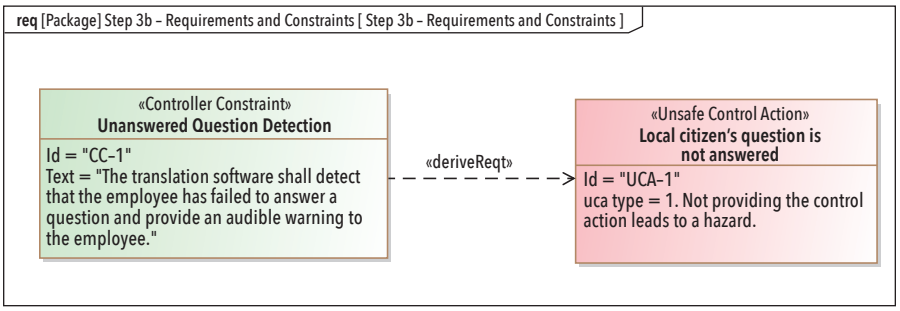


Figure 16. Step 3b: controller constraints

more formal approach. See Thomas (2024).

Roughly speaking, the four classes of loss scenario (Figure 17) entail:

- Class 1 – The controller's process model is not doing the right thing. As John Thomas explains, this situation most commonly arises when the controller has a mistaken belief about how the world works, what the current state is, or what needs to be done next.
- Class 2 – Incorrect or insufficient inputs. Previously, we had already identified that our base system design was not providing enough feedback to the employee in some situations. In Step 4, we would systematically look for more situations of this kind.
- Class 3 – The controller acted correctly, but the process did not respond, usually because something in the path from the controller to the controlled process failed.
- Class 4 – The controller did the right thing, and the control action arrived at the controlled process correctly, but an internal problem in the controlled process caused a problem.

John Thomas recommends setting up a discussion framework with the known control actions and unsafe control actions laid out in a sort of grid and ready for discussion and then using that material to drive a working session with the subject matter experts.

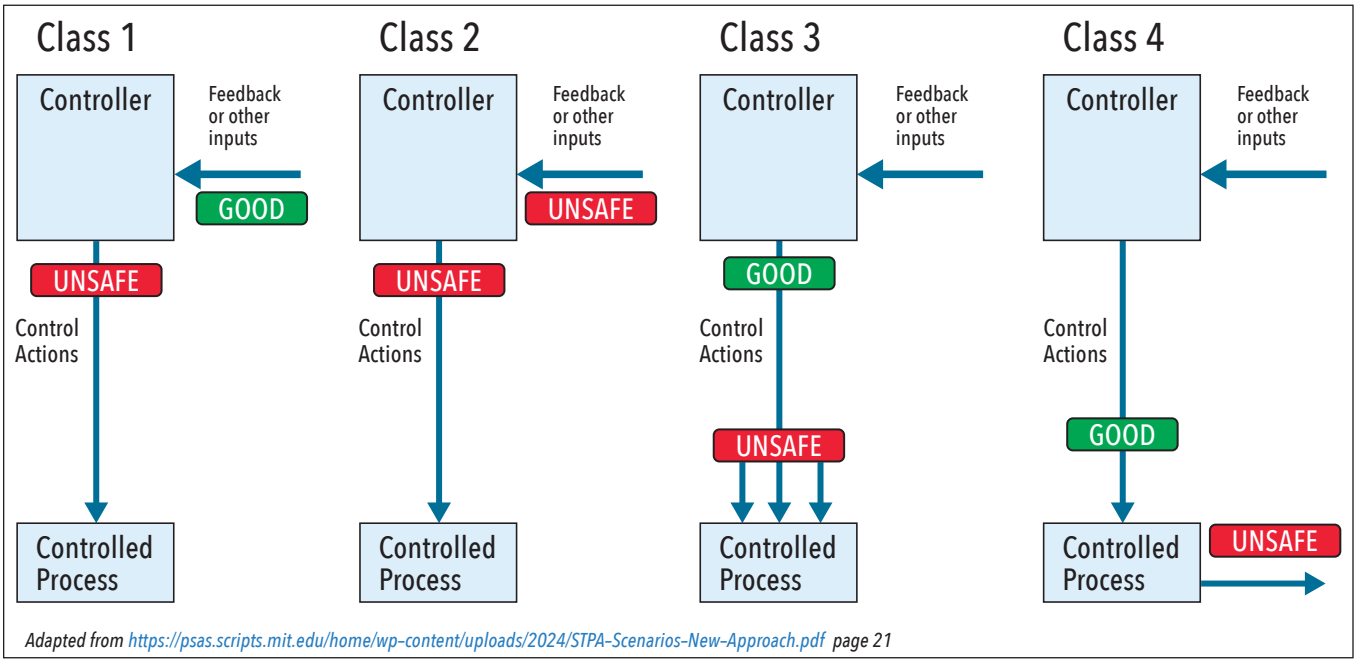


Figure 17. Four classes of loss scenarios

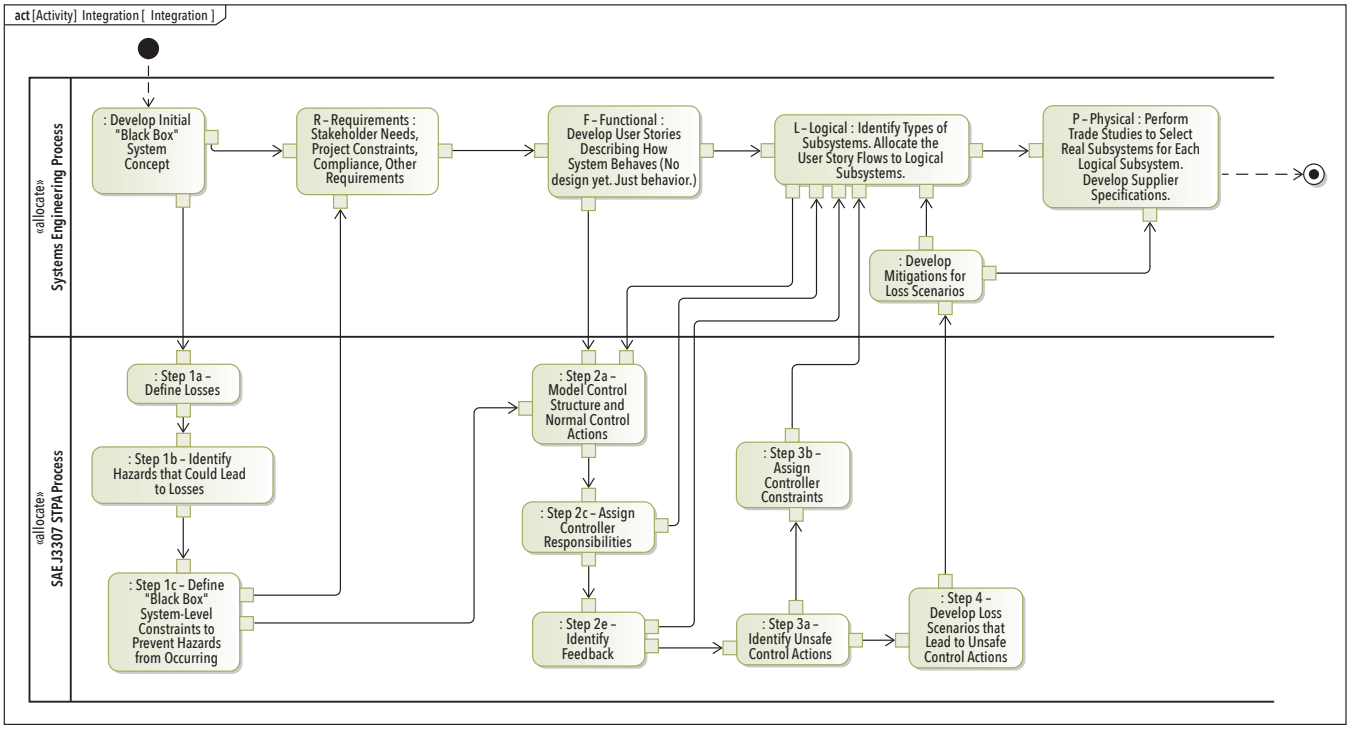


Figure 18. Overview of the integration of STPA with a systems engineering process

INTEGRATION OF STPA INTO A SYSTEMS ENGINEERING FLOW

In practical terms, how do we integrate STPA into a traditional systems engineering process?

Figure 18 presents and overview of the integration of the SAE J3307 STPA process with the RFLP systems engineering process. We can also look at this integration in terms of what specifically flows back and forth (Figure 19).

1. STPA cannot begin until we have a solid concept for the purpose of the system and some idea of the context the system will be operating in.
2. Once we have the purpose defined, we will usually be able to start with STPA Step 1 and develop the STPA losses, hazards, and system constraints. These should be kept concise enough that they can be directly reviewed with the stakeholders.
3. In order to develop the control structure in STPA Step 2, we will need some definition of the expected flows (usually from one or more user stories) and an idea of the logical structure of the system. They STPA team may chose to model these slightly differently than the systems engineering team. However, the STPA team will need to take care that alternative abstractions are logically equivalent and don't distort the actual design intent of the larger engineering team (Figure 20).

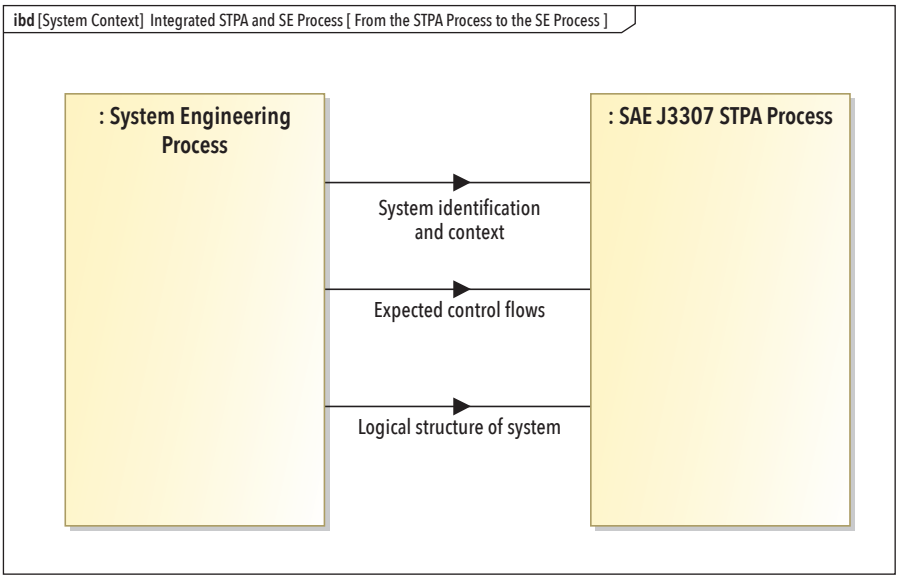


Figure 19. Flows from the systems engineering process to the STPA process

In the opposite direction, the STPA process yields three levels of formal engineering requirements:

- System-level constraints – are hazard avoidance goals that apply at the top level of the system.
- Controller responsibilities – are the allocation of portions of the system-level constraints to specific controllers.
- Controller constraints – are requirements to block specific unsafe control actions.

The STPA process will also yield two forms of design problem input.

- Feedback needs – Investigation of controller responsibilities and controller constraints will identify missing feedback that the controller will need to meet the requirements. However, the decision on exactly how to implement the feedback will usually be outside the scope of the STPA team.
- Loss scenarios for mitigation – These are the result of the deeper investigation

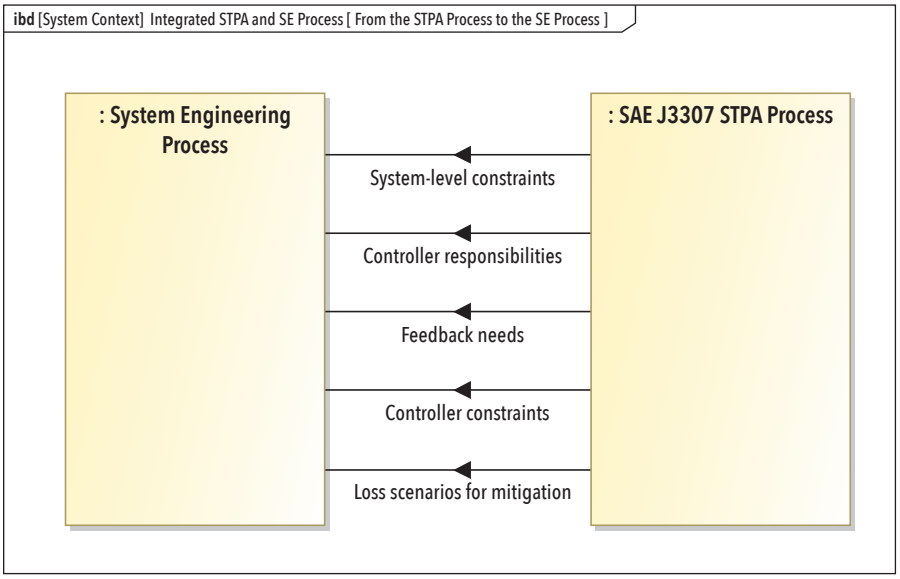


Figure 20. Flows from the STPA process to the systems engineering process

in Step 4. In many cases, these scenarios will include situations in which no single component has failed, but hazardous situations occur anyway. Mitigating these sorts of scenarios may require fundamental reassessment of the system architecture. Identifying the scenarios is a major service provided by the STPA team. Deciding exactly how to mitigate the scenarios will usually be outside the scope of the STPA team.

CONCLUSIONS

This article has demonstrated how STPA can be effectively applied to loss-driven engineering and seamlessly integrated into a model-based systems engineering workflow. Through a detailed example case, we explored a specialized personal communication device employing advanced software capabilities to support employees of a company operating in a risky environment in their efforts to build natural and supporting relationships with the local citizens. STPA was employed to examine possible losses and hazards in the design and use of such a device. Finally, we took a closer look at how the recently released SAE J3307 normative standard for STPA could be implemented in a SysML tool and integrated with a RFLP systems engineering process.

By applying STPA within a model-based systems engineering framework, we systematically identified these risks and developed additional system requirements to mitigate them—ensuring that the human communication and situational awareness goals of the system remain intact while maintaining acceptable levels of security protection. As model-based techniques continue to evolve within systems engineering, STPA provides a powerful, structured approach to integrating safety and security considerations into complex system design and allowing companies to develop sophisticate technology and human systems that can handle challenges and persevere in inherently risky environments. ■

REFERENCES

- Baughey, K. 2011. *Functional and Logical Structures: A Systems Engineering Approach. SAE Technical Papers on CD-ROM/SAE Technical Paper Series. https://doi.org/10.4271/2011-01-0517*.
- SAE. 2025. SAE J3307_202503: *System Theoretic Process Analysis (STPA) Standard for All Industries–SAE International*. Sae.org. https://www.sae.org/standards/content/j3307_202503/.
- Leveson, N., and J. Thomas. 2018 (March 1). STPA Handbook [Review of STPA Handbook]. https://psas.scripts.mit.edu/home/get_file.php?name=STPA_Handbook.pdf.
- Thomas, J. (n.d.). "STPA Step 4 Building Scenarios: A Formal Scenario Approach." Retrieved May 1, 2025, from https://psas.scripts.mit.edu/home/wp-content/uploads/2024/STPA-Scenarios-New-Approach.pdf.

ABOUT THE AUTHOR

David Hetherington is a principal at System Strategy, Inc. He is an engineering leader with over 40 years of experience in complex systems and organizational transformation. David helps clients with individual skills in model-based systems engineering as well as with team skills for the broader digital engineering organizational transformation. David is a member of the SAE G33 committee working on updates to the SAE GEIA-HB-649 – Configuration Management Standard Implementation Guide. David is also a member of the SAE STPA Task Force contributing to the forthcoming SAE J3187-4 recommended practices for applying STPA to system security. David speaks Japanese and German fluently and lives in Austin, Texas.

How Security Needs Systems Engineering

Mark W. Winstead, mark@markwinstead.net

■ **ABSTRACT**
Peter Neumann once noted that complex systems are not like snapping Lego pieces together, rather each piece added can transmogrify its modular interface and upset the existing structure. The effect for security can be a system weaker than its weakest link – moreover, addressing a security concern can disrupt achieving other emergent properties (e.g., safety). The article addresses these challenges by casting security as a system problem, where the security engineering must not be done stove piped from system engineering. The discussion within also addresses the role of systems thinking and the need for evidence-based assurance overseen by systems engineering.

■ **KEYWORDS:** Security, resilience, complexity, evaluation, assurance

INTRODUCTION

In recent decades, common practice often has systems engineering delegate security *(freedom from those conditions that can cause the loss of assets with unacceptable consequences* (Ross et al. 2022) to specialists, but historically systems engineering was once acknowledged as necessary for security. That is, effectively achieving security needs systems engineering perspectives, concepts, and activities.

In a seminal report for security (Neumann 2004), provable secure operating systems (PSOS) and trustworthy architecture pioneer Peter Neumann noted security is a system problem and that a system may be even weaker than its weakest link and that a system should be evaluatable and evaluated. Together, this suggests a conclusion also made by Neumann that a system's security engineering must not be done independently from the system's total engineering.

This article reexamines Neumann's perspective, revalidating that systems engineering has a critical role in achieving system security, going further to examine how security for a system CANNOT be done independently from the total engineering for a system and in fact depends on the systems engineering.

This necessity conclusion comes from examining the need to control behavior and outcomes for a system to be considered adequately secure. This is derived from observations (Ross et al. 2022) that to consider a system secure the system must:

- Deliver on its purpose within missions executed in foreseeable operating conditions. That is, the system must deliver its required capability despite intentional and unintentional forms of negative influences (i.e., adversity) both known and reasonably conceivable. This is sometimes referred to as being resilient.
- Enforce constraints to ensure that only the desired behaviors and outcomes associated with the required system capability are realized while delivering mission.
- Enforce a set of rules defining authorized entity-to-entity interactions and operations that are allowed to occur while delivering mission and ensuring only desired behaviors and outcomes.

Note that delivering mission is a systems engineering responsibility, and that to achieve the necessary enforcement of constraints and rules, system capability, desired behaviors, desired outcomes, and the set of rules must be defined unambiguously. This definition is a critical systems engineering role through its existing needs elicitation and requirements definition roles.

Additionally, if this control is not achieved with a justified confidence that comes from evidence, then the resulting deficit in assurance of the control functionality translates to risk. Evidence is generated through systems analysis, verification, and validation, all systems engineering responsibilities.

Finally, this article closes with discussing the need for stakeholder engagement. Security is not for security's sake, rather is an enabler to system objectives as well as a trade space item – need is for adequate security, not perfect security. Systems engineering collaborates with stakeholders to determine the integrated set of needs, prioritized to inform engineering decision making and trade appropriately considering security needs and concerns.

SECURITY IS A SYSTEM PROBLEM

Peter Neumann notes (Neumann 2004):

> *Ideally, we would like the development of complex hardware/software systems to be like snapping Lego pieces together! Instead, we have a situation in which each component piece can transmogrify its modular interface and its physical appearance — thereby continually disrupting the existing structure.*

John Thomas echoed Neuman when he said that *security behaviors are system behaviors, not system element attributes* (Thomas 2013). That is, security is an emergent system property that results from how systems engineering arranges the elements to interact within the system's context.

Moreover, simply using "secure" elements does not achieve system security. Security as a system behavior necessitates knowledge about how the element is intended to interact with other elements across the system states and modes to judge the element's security. That is, to judge an element to be secure can only be meaningfully determined and understood within a system's context.

Thus, security is a system problem, requiring systems thinking to understand and manage the behavior of the whole, including understanding and managing the conditions that may lead to loss. These conditions may be internal to a system, external to a system, or a combination. Such conditions involve all elements of the system and environment, requiring applying a socio-technical interpretation of system control theory.

A System May Be Weaker than its Weakest Link

Both security and insecurity are emergent properties. That is, a system's composition may produce secure and insecure behaviors or outcomes, even when the elements within the composite may be reused from other contexts where the elements were evaluated as acceptably secure.

Consequently, a system may be weaker than its weakest link since a system may display unspecified behaviors and outcomes – unintended behaviors and outcomes that may be harmful. Engineering must give specific attention to analysis to expose such unspecified behavior and outcomes and subject them to rigorous evaluation. Without such efforts, the result is a lack of a basis to judge the composed system as secure or insecure. This absence is a form of uncertainty, which is still another way a system may be weaker than its weakest link.

Systems engineering that leverages systems thinking aids both understanding and managing the behavior of the whole, thus managing those undesired emergent behaviors that make a system insecure.

Evaluatable and Evaluated

Evaluatable refers to conditions or states that enable rigorous and conclusive evaluation while *evaluated* refers to the end state of analytic activity that produces meaningful conclusive results. That is, *evaluatable* is a precondition for engineers to rigorously *evaluate* to produce justified results.

Confidence in realizing the characteristics of a secure system listed earlier must be grounded in evidence that justifies the confidence, including evidence generated by evaluations. Confidence is needed:

- In the completeness of the defined intended behaviors and outcomes (i.e., the set of requirements)
- In the correctness of the implementation that achieves the intended behaviors and outcomes, and
- That any unspecified behavior does not cause harm.

Evaluatable is one of the four essential design criteria for a security mechanism (Uchenick and Vanfleet 2005), along non-bypassability (can't go around it), always on, and tamperproof. Evaluatable requires mechanisms to be sufficiently small and simple enough to enable rigorous proof of correctness through mathematical verification. An entire system may not require being rigorously proven correct through mathematical verification, but engineering should sufficiently analyze a system to justify conclusions about claims of meeting the system's security needs with sufficient confidence. The analytic rigor for any one systems engineering artifact needs to be commensurate with the loss that may happen in the event of associated failures or compromise. Consequently, principles of *commensurate rigor, commensurate trustworthiness*, and *substantiated trustworthiness* inform needed evaluation and the precondition of a system or system element being evaluatable (Ross et al. 2022).

Moreover, the greater the consequence of loss and the confidence needed; the more engineering must generate and otherwise obtain the relevant evidence needed across the entirety of the system life cycle. A claims-oriented structured argument approach to assurance (i.e., assurance case) within systems engineering serves to address safety and security concerns (e.g., addressing potential gaps in derived requirements) and properly identify evidence to generate (ISO/IEC/IEEE 2023).

Impeding evaluability is complexity. Complexity is not always avoidable but can be managed to enable sufficient evaluations. Systems engineering manages both the complexity associated with the conduct of engineering activities and complexity with the interactions of engineering disciplines.

Security Engineering as Part of Systems Engineering

A significant systems engineering role is to optimize across all objectives for the system, including security and resilience. This role requires integrating engineering and scientific disciplines, including security disciplines, and managing interactions among the disciplines. For example, systems engineering uses complexity science to manage system complexity to manage emergence and to enable evaluation in a manner that confidence can be sufficiently established. Systems engineering balances the evaluation comprehensiveness with objectives associated with affordability and schedule.

Systems engineering also has a fundamental responsibility to precisely understand how elements interact and any resulting emergent behaviors and outcomes. The systems engineering team performs activities and tasks within a system's life cycle processes with the intent to continuously deliver capability from a combination of elements and interactions, thus addressing security as a system problem and managing emergence using systems thinking, often through a socio-technical interpretation of system control theory (Young and Leveson 2014).

The value of integrating security within systems engineering was once well-established. In 1989, the United States Department of Defense issued Military Standard 1785 (Department of Defense 1995) reflecting security integrated into systems engineering, republishing it as a handbook in 1992. In 2002 the NSA's since-retired Information Assurance Technical Framework (National Security Agency 2002) presented security engineering as part of system engineering.

Recent years have seen a return to realizing the need for a security role and perspective within systems engineering. For example, in 2022 alone:

- *Systems Engineering Vision 2035* (INCOSE 2021) recognized that security should be a foundational perspective for system design and
- NIST SP 800-160 Volume 1 Revision 1 described its intent as advancing systems engineering in developing trustworthy systems for contested operational environments.

CONTROLLING BEHAVIOR AND OUTCOMES

Systems engineering achieves secure system function by engineering to achieve

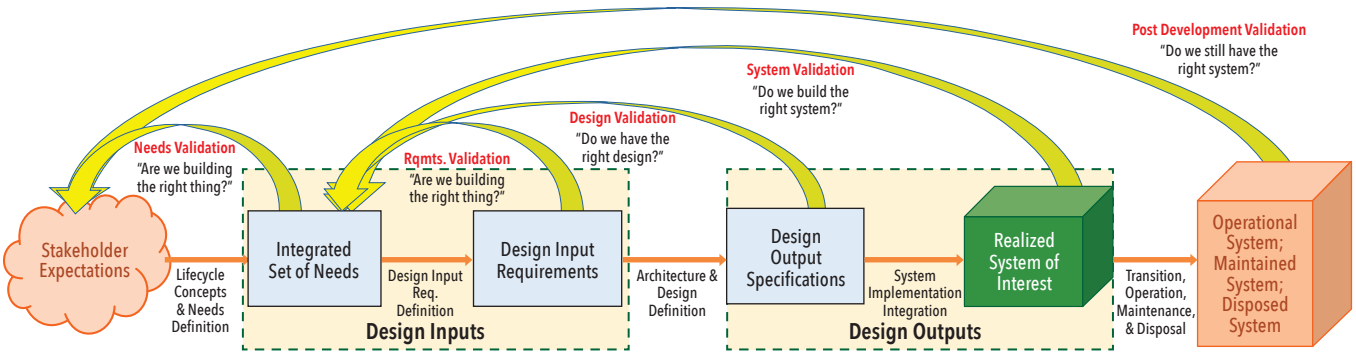


Figure 1. Relationships of the systems engineering engagements with stakeholders, adapted from (Wheatcraft et al. 2022)

only intended and authorized behaviors and outcomes and avoid unacceptable loss. This functional interpretation of security drives the need to maintain proper behavior within the system states and conditions, avoiding states and conditions that are undesirable.

Moreover, the "essence of risk management lies in maximizing the areas where we have some control over the outcome while minimizing the areas where we have absolutely no control over the outcome and the linkage between effect and cause is hidden from us" (Bernstein 1998). To manage what is often referred to as "cyber risk," this observation points to maximizing where we have control on behaviors and outcomes and minimize where we do not. While the motivation for security a response to known attacks, an approach to system design that places focus on the response to attacks will not fully maximize the potential for effective control and will not minimize those conditions with hidden linkage between cause and effect.

Effective control requires the ability to detect and respond to any failure. Failure is, and results in, undesired and/or unintended behaviors and outcomes that a malicious adversary may exploit or try to cause. Thus, understanding all failure (intentional and unintentional) and its consequences is essential to achieve secure system function. No individual discipline alone can identify all failures and consequences – it requires all disciplines. While individual disciplines have expertise in their domain (e.g., mechanical engineers know best how mechanical failures may occur, software engineers know best how software failures may occur), the system level ramifications of failure require identifying failures and their effects on establishing and maintaining desired behaviors, states, and conditions, and managing conditions that lead to failure. As a system level issue, all disciplines and stakeholders must contribute.

The required collaboration to effectively manage and integrate engineering and scientific disciplines is a systems engineering role. Acquisition and product development efforts can use systems engineering to manage the disciplines to avoid susceptibility, vulnerability, and hazard to the extent practical (see below) and control those not avoided, through systems thinking and a socio-technical interpretation of system control theory. That is, no single discipline (e.g., the 'cyber' disciplines) can account for all system level security concerns without collaboration across all relevant disciplines.

Success in controlling behaviors and outcomes require clearly defining the intended behaviors and outcomes, assurance in efforts to control the behaviors and outcomes, and design concepts captured in the *inherently secure design* concepts.

Assurance

Evidence provides grounds for the justified confidence in claims about system properties (ISO/IEC/IEEE 2019) (i.e., evidence for trust judgments). The strength of assurance is a function of the evidence's relevance, credibility, and accuracy, as well as the ability to craft valid, logical, and compelling arguments that use the evidence to substantiate stated claims. Assurance is dynamic, changing as aspects of the system, the environment, and the intended use change, and as history and observations are collected. Therefore, systems engineering must continuously maintain assurance to accurately represent the system at all points in time when stakeholders need updated assurance-related judgments.

The results and rigorous conduct of systems engineering with support from other engineering activities provide the strongest objective evidence for assurance. Especially important for assurance is evidence that reflects the rigorous conduct of system analysis, verification, and validation as well as the evidence produced by the results of those activities.

Assurance in effectively achieving system objectives increases certainty. Assurance deficiencies (i.e., where the assurance achieved is less than the assurance stakeholders seek) present uncertainty and the potential for risk. Assurance considerations are a significant factor in all system engineering activities, to ensure that the results of those activities are convincing when subjected to scrutiny.

Inherently Secure Design

A system, enterprise, or organization may lose control over system behaviors and outcomes through susceptibility (an inability to avoid being "hit"), vulnerability (an inability to withstand a "hit), and hazard (conditions with potential to leading to suffering loss). Susceptibilities include exposure to potential danger like a connection to the internet and vulnerabilities include weaknesses such as a lack of checks of validity to software module inputs. Hazards include conditions within and outside a system, such as adverse weather for a sailboat.

An inherently secure system avoids susceptibility, vulnerability, and hazard to the extent practical as a direct by-product of the design. The "inherently secure basis" to such a design reduces the number of engineered functions necessary to exercise control over the susceptibility, vulnerability, and hazard that could not be avoided. Reducing engineered functions results in a simpler design and fewer security functions to protect from both attacks and their own erroneous behaviors and failures. The author discussed inherently secure design concepts more extensively in Winstead (2024).

Inherently secure design proactively addresses maximizing what is controlled and minimizing what is not controlled. Design activities should achieve inherently security design as a *direct by-product*, as part of larger engineering activities overseen by systems engineering, not as a part "cyber" activity which too typically assesses risk in a design and mitigates that risk by only building in countermeasures (e.g., security controls) rather than striving to eliminate the risk.

STAKEHOLDER ENGAGEMENT

Stakeholders collectively have multiple needs, goals, objectives, priorities, and constraints, not all of which are security focused. Systems engineering continually engages in technical exchanges with stakeholders to acquire their inputs, to manage their technical expectations, and to validate that the system being realized meets expectations (i.e., is "fit for purpose") as illustrated in part by Figure 1.

This engagement includes eliciting stakeholders' comprehensive needs. Specific for security, this includes eliciting loss concerns to inform a comprehensive expression of stakeholder protection needs (Ross et al. 2022; Scheuer and Wilson 2024). Systems engineers collaborate with stakeholders to integrate these needs into the set of needs, including the prioritization of needs to better inform engineering decision making and trades.

A security relevant goal of the integrated set of needs is to ensure that security within the system is not "security for security's sake." Instead, security needs are to enable the achievement of stakeholder needs in a secure manner. The engagement with stakeholders continues throughout the life cycle to validate needs and ensure engineering outcomes align with needs, and the realized system meets the needs.

CONCLUSION

System behaviors and outcomes inform the judgement of whether a system is secure or not secure. These behaviors and outcomes are an emergent outcome produced by the interactions across combinations of element behaviors and outcomes. Consequently, security is an emergent property and thus a system problem.

This, combined with observations made within the paper of systems engineering's role in engineering a system for sufficient control over behaviors and outcomes and building evidence for assurance of that control, leads one to conclude Peter Neuman was incomplete on one point in his 2004 report. It is not just the case that the security engineering of a system must not be done independently from the total engineering of the system (i.e., the systems engineering and engineering disciplines managed under systems engineering), rather than a system's security engineering is intertwined inseparably from the total engineering of the system.

Subsequently, it is not enough to embed "cyber" expertise into teams working with systems engineering. The need is to see security as foundational in systems engineering as system performance is today. The systems engineering discipline needs to grow to be even more transdisciplinary, especially as it relates to security. By extension, all engineering disciplines need to expand thinking to systems and products operating in contested environments, including often engineering in contested development environments.

Moreover, stakeholders need systems engineering to ensure adequate security while meeting the system's purpose within missions' objectives, and not as an exercise in "security for security's sake." Systems engineering's inclusion and its leadership are needed in establishing adequate security appropriate for a system's purpose (Ross 2024) and its associated capability requirements, so a system perseveres to deliver required capability. Stakeholders need systems engineers to perform the proper trades and other decisions among all needs, goals, and objectives. These trades and decisions must support requirements and design activities that ensure inherently secure thinking.

In short, systems engineering must step up to take responsibility for security and not delegate it solely to specialists. ■

REFERENCES

- Bernstein, P. 1998. *Against the Gods: The Remarkable Story of Risk*. s.l.:Wiley.
- Boudra Jr, P. 1993. *Report on rules of system composition: Principles of secure system design. I9 Technical Report*. National Security Agency, Information Systems Security Organization, Office of Infosec Systems Engineering, Fort Meade, US-MD.
- Department of Defense. 1995. MIL-HDBK-1785: Department of Defense Handbook - System Security Engineering Program Management Requirements. Washington, US-DC.
- INCOSE. 2021. Systems Engineering Vision 2035. [Online] Available at: https://violin-strawberry-9kms.squarespace.com/ [Accessed 11 April 2025].
- INCOSE. 2022. *Systems Engineering Principles*, San Diego, US-CA.
- ISO/IEC/IEEE. 2019. ISO/IEC/IEEE 15026:2019 Systems and Software Engineering–Systems and Software Assurance–Part 1: Concepts and Vocabulary. Geneva, Switzerland.
- ISO/IEC/IEEE. 2023. ISO/IEC/IEEE 15288:2023 Systems and software engineering – System life cycle processes. Geneva, Switzerland.
- National Security Agency. 2002. *Information Assurance Technical Framework*. [Online] Available at: https://ntrl.ntis.gov/NTRL/dashboard/searchResults/titleDetail/ADA606355.xhtml [Accessed 11 April 2025].
- Neumann, P. G. 2004. *Principled Assuredly Trustworthy Composable Architectures*, Menlo Park, US-CA: SRI International.
- Ross, R. 2024. *Next Generation Mission-Based Security for Systems Engineers*. McLean US-VA: NIST.
- Ross, R., M. Winstead, and M. McEvilley. 2022. NIST SP 800-160 Volume 1 Revision 1 Engineering Trustworthy Secure Systems, Gaithersburg, US=MD: National Institute of Standards and Technology.
- Wilson, B., and A. Scheuer. 2024. *Guide to Security Needs and Requirements*, San Diego, US-CA: INCOSE.
- The MITRE Corporation. 2022. *Security and Resilience Interpretation 1.0*, s.l.: OUSD(R&E) Systems Security Directorate.
- Thomas, J. A. 2013. *INSIGHT* July, 16 (2): 3.
- Uchenick, G. M., and W. M. Vanfleet. 2005. "Multiple independent levels of safety and security: high assurance architecture for MSLS/MLS." *MILCOM 2005-2005 IEEE Military Communications Conference*, Volume 1, pp. 610-614.
- Wheatcraft, L., T. Katz, M. Ryan, and R. B. Wolfgang. 2022. *INCOSE Needs and Requirements Manual: Needs, Requirements, Verfication, Validation Across the Lifecycle*. San Diego, US-CA: INCOSE.
- Winstead, M. 2024. "Secure Design: A Principled Approach for Systems Engineers." *Proceedings of the INCOSE International Symposium*, July 2025, Dublin, Ireland: Wiley.
- Young, W., and N. G. Leveson. 2014. "An Integrated Approach to Safety and Security Based on Systems Theory." *Communications of the ACM*, February, 57 (2): 31-35.

ABOUT THE AUTHOR

Mark Winstead is chief engineer for systems security at The MITRE Corporation and an instructor for Cal Tech's Center for Technology and Management Education (CTME). He is an author of *NIST Special Publication 800-160 Volume 1 Revision 1 Engineering Trustworthy Secure Systems*. With INCOSE, Mark is co-chair of the Systems Security Working Group as well as co-chair of the Loss-Driven Systems Engineering Project.

Illuminating Systems Security Through Case Studies – Much More than Controls

Beth Wilson, wilsondrbeth@aol.com

■ **ABSTRACT**
While systems security is a quality attribute (previously referred to as specialty engineering), learning systems security is essential for all systems engineers. Learning about system security can be a challenge especially when the focus is on security controls or admiring attack vectors. Case studies are a powerful way to see the real-world application of complex concepts. Reviewing cyber-attack case studies provides a captivating approach to examine security challenges and failures holistically using systems thinking, consider the technical concerns, business decisions, and human behaviors that made the attack possible, and explore systems security concepts from a systems engineering perspective.

INTRODUCTION

The INCOSE Systems Security Working Group (SSWG) has been working on strategic concepts related to security in the Future of Systems Engineering (FuSE) (Dove 2022) defining objectives and strategies to develop and evolve practices. Figure 1 highlights the strategic concept "security proficiency in the systems engineering team" that focuses on improving the interaction between systems engineering practitioners and systems security practitioners and increasing the systems engineering practitioner knowledge of systems security concepts.

Even when a systems engineering practitioner is inspired to learn more about systems security, it is difficult to know how to proceed. An online search for learning opportunities in systems security will lead to training programs to support information technology certifications and network cybersecurity topics. Systems security involves much more than networks between system elements.

Focusing on individual cybersecurity topics such as encryption or access control

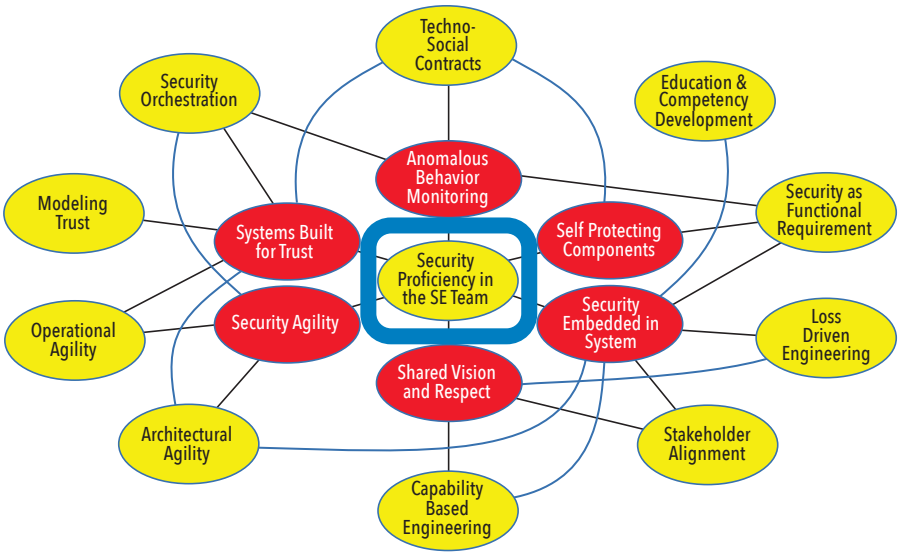


Figure 1. The strategic concept "security proficiency in the systems engineering team" focuses on improving the interaction between systems engineering practitioners and systems security practitioners and increasing the systems engineering practitioner knowledge of systems security concepts

misses the bigger picture of how to design a system that is secure. The National Institute of Standards and Technology (NIST) Special Publication (SP) 800-53 "Security and Privacy Controls for Information Systems and Organizations" (NIST 2020) provides a catalog of security tactics organized into control families. While it is a helpful taxonomy of approaches and provides consistent terminology, in isolation it promotes a compliance mental model that is not helpful in designing secure systems. Referring to role-based access control as AC-3(7) does not turn a systems engineering practitioner into a systems security practitioner any more than referring to an interface diagram as SV-1 (Systems View 1) turns a systems engineering practitioner into a systems architect.

Improving security proficiency in the systems engineering team means having systems engineering practitioners that think holistically about systems security as a necessary part of system design. Instead of perseverating on the controls in NIST SP 800-53, systems engineering practitioners explore the principles of trustworthy design in NIST 800-160 (Ross 2022). Systems engineering practitioners understand enough about systems security to make it part of their design and to know when to call on a systems security practitioner to supplement their knowledge and complement their skills. Systems engineering practitioners have basic knowledge so that they can communicate more effectively with systems security practitioners. Systems engineering practitioners apply systems thinking to systems security so that the resulting system can prepare itself for threats, defend against threats, and recover essential capability when attacked to achieve mission success.

CHALLENGES IN LEARNING ABOUT SYSTEMS SECURITY

In trying to learn about systems security, the challenge is in what topics to explore and how much detail is necessary in these topics. There are obvious topics in describing different kind of threats such as malware, denial of service, and insider threats. How much detail is necessary to understand a class of threats and how to protect against it? There are vulnerabilities that can be mitigated with basic cyber hygiene, but we need to learn more than "don't click on attachments." While it is important to learn about defense in depth architectures and techniques such as firewalls, intrusion prevention, and intrusion detection, it is also important to understand the design principles that use these techniques.

The systems engineering practitioner needs to learn that systems engineering is responsible for systems security, and it cannot simply be a quality characteristic captured as non-functional requirements or added later. This means understanding the systems security tasks related to systems engineering activities throughout the system lifecycle.

The systems engineering practitioner needs to understand systems security terms and concepts. They need to understand systems security principles for trustworthy secure design and vigilant system use. They need to understand the difference between security strategies, tactics, and techniques and apply systems thinking to understand the system holistically.

The systems engineering practitioner needs to learn how to perform needs-oriented, loss-driven, capability-based analysis to understand what parts of the system most need to be protected and what system capabilities most need to prevail in a potentially degraded state. This means understanding which mission threads are critical and identifying system capabilities in the form of functional security requirements to mitigate loss scenarios.

The systems engineering practitioner needs to learn how to protect the system from attack and how to respond to an attack. This means addressing both cyber security and cyber resiliency. They need to know how to build an assurance case to provide evidence of secure and resilient system capability. They need to understand what test and evaluation approaches can verify security requirements and validate security needs.

Security proficiency in the systems engineering team means that the systems engineering practitioners have enough systems security knowledge to be able to design a system that will achieve mission success in the face of adversity. Obtaining an awareness level proficiency for selected security concepts will provide the systems engineering practitioner with important vocabulary and an appreciation for system security concerns. The challenge is in putting theory into practice, which is why case studies can be a beneficial way to learn system security concepts.

BENEFITS OF CASE STUDIES

Case studies provide a real-world application of complex concepts and help us learn from other's failures. Using a case study of a system security failure can show how vulnerabilities were exploited with real consequences. The story of what the Stuxnet malware was able to do in the Natanz Fuel Enrichment Plant puts malware theory into practice and shows how cyber-physical attacks can happen.

Case studies bring concept understanding through storytelling. The topic of insider threat brings about thoughts of monitoring employees. The story of the wastewater supervisory control and data acquisition (SCADA) attack by a disgruntled former employee brings the insider threat concepts to life in a meaningful way.

Case studies provide knowledge through problem solving. Looking at real cases that have happened allows us to explore what vulnerabilities existed and how the attack could have been prevented. The story of the watering hole attack against energy firms in 84 countries shows supply chain vulnerabilities against the backdrop of cyber espionage.

Case studies encourage critical thinking to explore the situation and understand the factors that led to the incident being described. It allows us to view the incident from multiple perspectives to identify vulnerabilities and how they came to be. Case studies provide a powerful tool to apply concepts and transition theory into practice.

SELECTED CASE STUDIES

Worcester Polytechnic Institute uses case studies in its graduate level class in systems security. Highlighted here are 10 case studies presented in chronological order to show increasing sophistication in the attacks and recognize that some lessons are never fully learned. The case studies are presented from the perspective of the victim to keep the focus on the vulnerabilities exploited in the attack and what the victim could have done to prevent the attack. The cases have changed since the course was first delivered in 2016, but the 10 cases being presented in 2025 are shown in Table 1.

Maroochy (Abrams 2008): In 2000, the Maroochy Water Services in Queensland, Australia was the target of an insider cyber-attack on its wastewater system. A subcontractor who installed the sewage control equipment was rejected for a job at Maroochy Water Service and became a disgruntled employee and insider threat. The impact of the resulting sewage spill was significant with over 800,000 liters of raw sewage spilled in parks, rivers, and a hotel. The attacker sent malicious radio commands to pumps disrupting the wastewater SCADA systems. The court proceedings against the attacker brought many of the details into open literature. What makes this an interesting case, besides demonstrating the insider threat, is that it is one of the earliest attacks on industrial control systems. In 2000, there were no access credentials or user authentication mechanisms for SCADA systems. There was no authentication required to access the network. The prevailing belief was that the system was secure because it was on a separate network and specialized skills were required to interact with it. This case

Table 1: Selected Case Studies

Year	Victim	Why Significant
2000	Maroochy	Insider Threat
2007	Estonia	First "cyber war" (DDoS)
2010	Natanz	First cyber-physical attack (malware)
2013	Target Corporation	Attack through subcontractor
2014	Energy Firms	Supply chain, cyber espionage
2014	JP Morgan Bank	Large security budget, financial domain
2015	Office of Personnel Management	PII, systemic vulnerabilities
2016	Dyn	IoT and commercial websites
2016	Ukraine Power Grid	Critical infrastructure, cyber resiliency
2020	Garmin	Ransomware

triggered efforts to secure SCADA systems that did not previously exist for critical infrastructure systems.

Estonia (Saleem 2009): In 2007, street riots protesting the government's decision to remove a World War II memorial turned into cyber riots. Over a period of 3 weeks, 128 unique distributed denial of service (DDoS) attacks were made on IP addresses inside Estonia. This was a country that had been a "paperless government" since 2001 and 95% of banking in Estonia was electronic in 2007. As an online society, they felt the impacts of the disruptions in their daily business activities in both the government and private sector. The Estonia attack was the first of its kind to show what happens when "patriot hacking" becomes cyber warfare. It brought about international awareness of the potential impact of DDoS.

Natanz (INCOSE SEH5E 2023): No list of security case studies would be complete without the Stuxnet malware attack on the Natanz Fuel Enrichment Plant. This was the first of its kind cyber physical attack where malicious software was introduced to attack hardware. By the time it was discovered in 2010, over 1,000 centrifuges were damaged. The incident increased awareness of the industrial control system vulnerabilities and cyber-physical security needs. When the INCOSE Systems Security working group was asked to provide a case study for the fourth edition of the *Systems Engineering Handbook*, this case study was the one selected and it was updated for the fifth edition.

Target (Shu 2017): In 2013 attackers launched a phishing campaign on one of Target's heating, ventilation, and air conditioning (HVAC) vendors. The stolen credentials provided access to Target's invoice and billing system, but once inside the attackers were able to install point-of-sale malware on the store terminals. Over 110 million customer records were compromised and over 40 million credit cards were stolen, costing Target $382 million to respond to and recover from the attack. What makes this case significant is that the attack vector was through a vendor. The HVAC vendor was a small company that had little security because they believed that a small company would escape the attention of cyber attackers. Target was not the only retailer that suffered a point-of-sale attack, but it was the largest at the time and became a tipping point for the payment card industry. The result was a push to move from the magnetic strip readers to chip technology.

Energy Firms (Nelson 2016): From 2013 to 2014, over 1,000 energy companies were compromised in 84 countries including the US and Europe. This case study represents a cyber-physical attack with the potential to impact critical infrastructure through the supply chain. The spear phishing campaign started in 2013 to deliver Trojan malware that compromised websites used to provide firmware updates for Industrial Control System components. The attackers were then able to extract information about the energy company operations and manufacturing processes. What makes this case significant is the recognition that the supply chain can be an attack vector. The attackers used industrial software suppliers to propagate malware to victims. Large manufacturers buy components and software from thousands of vendors. This case brought awareness to supply chain as a cyber vulnerability. Another interesting part of this case is that unlike the Stuxnet malware that was a cyber physical attack to inflict damage, this malware was intended for industrial espionage.

JP Morgan Bank (Jeng 2015): In 2014 a malware attack on an employee's personal computer allowed attackers to gain login credentials. This shouldn't have allowed the attackers access to the JP Morgan network, but the two-factor authentication was not enabled on an internal server. The attackers moved around the JP Morgan network undiscovered for 2 months compromising 83M customer records. One of the things that makes this case study significant is that the victim of this attack was serious about cyber security, spending a lot of money on cyber security with many employees dedicated to security. This also sparked conversation about vulnerabilities in the financial industry. The discussion shifted from preventing attacks to detecting and minimizing damage from attacks.

Office of Personnel Management (Congressional Report 2016): From 2014 to 2015, attackers were able to obtain credentials and infiltrate the US Office of Personnel Management (OPM) networks that contained sensitive information about federal employees and background investigations for security clearances. Over 21.5 million records with personally identifiable information were stolen from the poorly secured servers. This case study represents a systemic weakness in system security. Sensitive information was stolen over several months while the OPM security team struggled to deal with the alerts. The investigation that followed revealed that OPM had conducted annual audits that identified serious security vulnerabilities going back as far as 2007. Annual audits were required. Mitigating the findings was not and OPM did not address these known vulnerabilities. This incident cost the taxpayers over $500M in credit monitoring and post-breach services. This was not the largest breach or the most expensive. What makes this significant is the type of data that was stolen. There was an immediate change in federal policies regarding retention of personally identifiable information.

Dyn (Mukundhan 2017): In 2016, a DDoS attack on Dyn made it difficult for people to access web pages on the Internet. The attackers infected over 100,000 Internet of Things (IoT) devices connected to the Internet with malware that they used to create a botnet. What makes this an important case is recognizing the vulnerability of IoT devices. At the time, there is no incentive for manufacturers to make the IoT devices or the connected systems secure. The consumers that bought these devices happily connected them to the Internet without changing the default password (if there was even a password to change).

Ukraine Power Grid (E-ISAC 2017): In 2015, hackers used phishing emails to install malware targeting industrial controls

systems and implemented a SCADA hijack, followed by a telephone DDoS attack on the call center causing a wide-spread power outage in Ukraine. In 2016, additional spear phishing campaigns allowed the hackers to install malware for another SCADA hijack in the capital region followed by a DDoS on the protective relays to delay power recovery. This case is significant because it represents one of the first publicly acknowledged incidents that resulted in power outages. It provided a look into cyber warfare of the future that targets a nation's critical infrastructure.

Garmin (Cyberint 2020): In 2020, Garmin products and services went offline when they fell victim to a ransomware attack. This is not the first ransomware attack, but it does represent a level of sophistication that moves ransomware from a target of opportunity to an advanced persistent threat. This case is also significant because it appears that the sizable ransom was paid. There are many lessons learned from this attack that other companies need to take in.

These cases represent a variety of attacks and domains. This short list of cases includes insider threat, malware, and denial of service attacks. The attack vectors are from within, from outside, using a subcontractor, through the supply chain, and opportunistic use of unprotected devices. The victims include small companies, large companies, governments with large and small security teams. The domains include commercial, financial, government, and energy domains. Some of the attacks are software-centric and some are cyber-physical.

We can consider different security topics with each case study. For example, the topic of insider threat is the primary focus of the Maroochy case. Denial of service is the threat vector for the Estonia case. Topics related to cyber physical systems, including techniques to protect industrial control systems, emerge with the Natanz case. How malware works and its countermeasures relates to many of the cases but is a key part of the Target case. Supply chain vulnerability relates to the energy firms case. The 2015 and 2016 attacks on the Ukraine power grid reinforce the cyber physical concepts and offer an opportunity to explore cyber resiliency as later waves of the attack targeted repair and recovery. Other security topics span these cases including defense in depth, security architecture, making a business case for systems security, security planning and response, data protection, intrusion detection, access control, user authentication, and security standards.

SYSTEMS THINKING FOR CASE STUDY INSIGHTS

Using systems thinking, we can see patterns across multiple cases and interactions that influence outcomes. None of the systems security concepts stand in isolation. These cases show that optimizing the security in one area still leaves the system vulnerable to attack. The JP Morgan case study shows that despite significant investment in security, leaving the two-factor authentication disabled on one internal server was enough for an adversary to get inside the network.

As system complexity grows, so do the vulnerabilities that expose the system to attack. These case studies show us that if the system is not architected to reduce attack vectors in its system elements, then emergent behavior at the system level introduces vulnerabilities that cannot be mitigated with even the most comprehensive security tactics at the hardware and software level.

Incident Contributors Overlap and Interact: For each case study we can consider technical concerns, business decisions, and human behavior in isolation. For each case study we can argue that technical concerns or business decisions or human behavior is the most significant and that rationale would be valid. Through the case studies we can see that while the attacks become more sophisticated, some of the mistakes that lead to vulnerabilities stay the same. It seems impossible to prevent every employee from clicking on a malicious attachment. It is not a matter of if an adversary will get in, but when.

Systems thinking shows us that there are overlaps between the incident contributors as shown in Figure 2. While it may be a business decision to provide awareness training, it is human behavior that implements the cyber hygiene principles provided in that training. While it may be a technical concern that the intrusion detection system installed to protect against attacks is not configured properly, it is human behavior that selected that configuration to reduce false alarms. While it may be a business decision not to keep up with operating system updates and patches, it is a technical contributor when the resulting vulnerabilities can be exploited by malware.

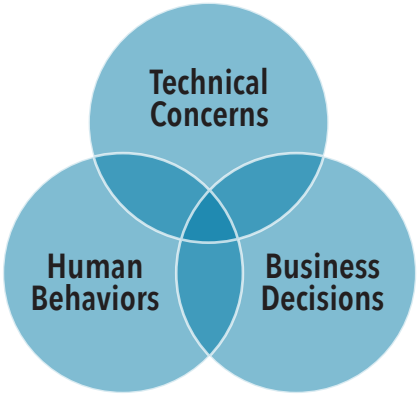


Figure 2. Case studies have overlapping and interacting incident contributors.

Attack vectors can be overlapping threats: Although some of the case studies have a single attack vector, we see that there are overlapping threats even in a simple attack with some examples shown in Figure 3. The attack may result from malware, but how did the malware get there? Was it installed by an insider? If so, the attack vector is both malware and

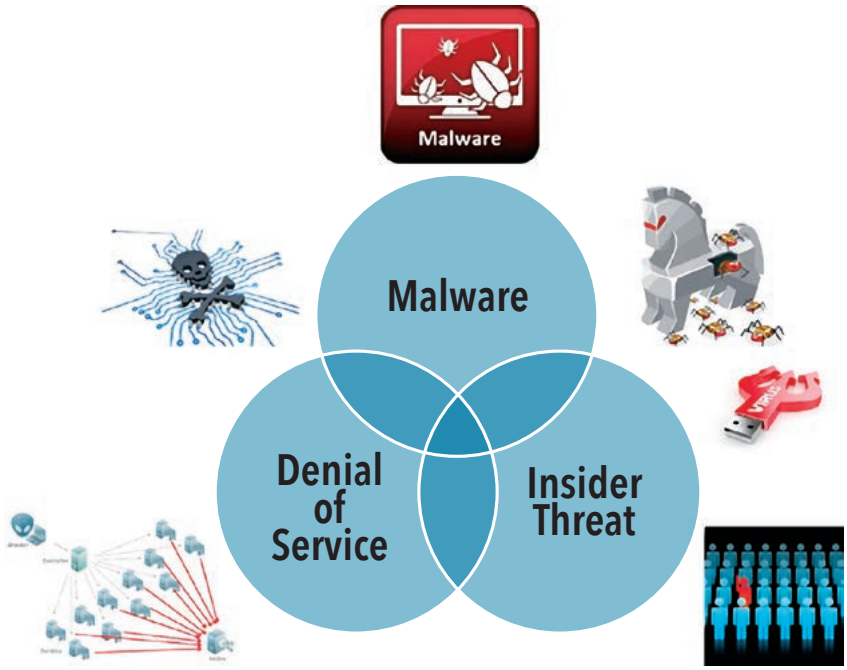


Figure 3. Attack vectors can be overlapping threats

insider threat. If a malware effect is to create bots that flood the victim's server, then the threat is both malware and denial-of-service. A phishing email that installs malware allows the attacker to steal credentials and makes the attacker an insider when they use those credentials. As we explore the real-world implications of these complex attack vectors, we begin to recognize the need to design a system to protect against multiple and possibly persistent threats.

Systems thinking shows us that we need to view interacting and evolving threats. Instead of focusing on protecting system elements against a specific kind of threat, systems thinking tells us we need to focus on the system capabilities necessary to mitigate losses created by any adversarial condition, intentional or unintentional.

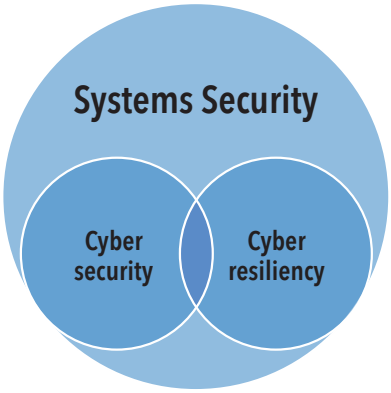


Figure 4. Systems security focus includes both security and resiliency

Figure 5. Security is a set of functional requirements to prepare the system for, defend against, and recover from adversity

Need holistic techniques to prepare, defend, and recover: Through these case studies, we see the need to design a system that can function under nominal and adverse conditions. Using the graphic in Figure 4, we can view systems security as both cyber security and cyber resiliency.

Systems thinking shows us that we need to view the system design holistically to consider the interaction between the system elements and the emergent behavior that can happen when the system is faced with an adversarial condition. We can perform loss-driven analysis to identify security needs for the system that describe what losses can be tolerated and still achieve mission success. Using these loss scenarios, we can define capabilities described as functional security requirements to prevent these losses. We can apply security concepts, approaches, and principles to design secure systems. We need to design systems to prepare themselves to keep the adversary out. We need to design systems to defend against an attack when an adversary gets in despite those preparations. We need to design systems to recover capability quickly. Figure 5 reminds us that security needs to be a set of functional requirements to prepare for, defend against, and recover from intentional and unintentional threats.

SUMMARY

Combining awareness of systems security topics with case study application of these topics can help systems engineers learn how to design secure systems and resilient systems. The case studies show interaction of concepts and challenges in preparing for, defending against, and recovering from adverse conditions. The holistic approach to system analysis promotes a needs-oriented, loss-driven, capability-based system design. This focus on essential capability can help us sustain systems security as threats evolve. ■

REFERENCES

- Dove R. 2022. "Setting Current Context for Security in the Future of Systems Engineering." *INSIGHT* 25 (2): 8-10.
- NIST Joint Task Force. 2020. NIST SP 80-53 Revision 5 Security and Privacy Controls for Information Systems and Organizations.
- Ross, R., M. McEvilley, and M. Winstead. 2022. NIST SP 800-160 Volume 1 Revision 1 Engineering Trustworthy Secure Systems.
- Abrams, M., and J. Weiss. 2008. "Malicious Control System Cyber Security Case Study — Maroochy Water Services, Australia" MITRE Technical Report. (https://www.mitre.org/sites/default/files/pdf/08_1145.pdf).
- Saleem, M. and J. Hassan. 2009. "'Cyber Warfare,' the truth in a real case." Project Report, Sweden. (https://www.ida.liu.se/~TDDD17/oldprojects/2009/projects/007.pdf).
- INCOSE. 2023. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, Fifth Edition. INCOSE-TP-2003-002-05.
- Shu, X., K. Tian, A. Ciambrone, and D. Yao. 2017. "Breaking the Target: An Analysis of Target Data Breach and Lessons Learned." ARXIV. (https://arxiv.org/pdf/1701.04940).
- Nelson, N. 2016. "The Impact of Dragonfly Malware on Industrial Control Systems." SANS Institute. (https://www.giac.org/paper/gicsp/724/impact-dragonfly-malware-industrial-control-systems/148912).
- Jeng, A. 2015. "Minimizing Damage from J.P. Morgan's Data Breach." SANS Institute. (https://www.giac.org/paper/gsec/36190/minimizing-damage-jp-morgans-data-breach/143120).
- Congressional Report. 2016. "The OPM Data Breach: How the Government Jeopardized Our National Security for More than a Generation." (https://oversight.house.gov/wp-content/uploads/2016/09/The-OPM-Data-Breach-How-the-Government-Jeopardized-Our-National-Security-for-More-than-a-Generation.pdf).
- Mukundhan, H. 2017. "Anatomy of an IOT DDoS Attack and Potential Policy Responses." *ISACA Journal* 5. (https://www.isaca.org/-/media/files/isacadp/project/isaca/articles/journal/2017/volume-5/anatomy-of-an-iot-ddos-attack-and-potential-policy-responses_joa_eng_0917.pdf).
- E-ISAC. 2017. "Modular ICS Malware." Electricity Information Sharing Analysis Center, Washington, US-DC. (https://assets.contentstack.io/v3/assets/blt36c2e63521272fdc/blt4f6cc0b6358c6883/607f235a6371c75a11ad9f5a/E-ISAC_SANS_Ukraine_DUC_6.pdf).
- Cyberint. 2020. "Evil Corp Wasted Locker Ransomware." Cyberint Report. (https://e.cyberint.com/hubfs/Cyberint_Evil%20Corp%20Wastedlocker%20Ransomware_Report.pdf).

ABOUT THE AUTHOR

> continued on page 60

JULY 2025
VOLUME 28/ ISSUE 3

When Malicious Actors Control Your Subsystems: A Systems Engineering Approach to Functional Perseverance

David Hetherington, david_hetherington@ieee.org; **Ivan Taylor,** ivan@policydynamics.ca

■ **ABSTRACT**
Security in modern engineered systems is not merely an added layer of protection but a prerequisite for system functionality. As systems engineers navigate the evolving security landscape, they must prioritize functional perseverance, the ability of a system to maintain operational integrity despite adversarial threats. This article examines a possible method for using system-theoretic process analysis (STPA) and system dynamics (SD) to enhance security-aware system engineering.

The approach shown is inspired by a 1982 paper called "The Byzantine Generals Problem" and is a peer-to-peer voting design that avoids single points of failure. In particular, we propose a system analysis and design approach that would allow the construction of a system capable of using peer-to-peer self-policing to detect an intruder that has already penetrated the security perimeter of the system and corrupted one or more of the subsystems. This article shows how STPA could inform the design of the peer-to-peer voting system and how SD could be used to examine the tradeoff of investments in redundancy versus the expected level of achieved resilience.

INTRODUCTION

Modern systems must prioritize resilience as a fundamental design principle to face adversarial threats. Functional perseverance—the ability of a system to sustain operational capability despite disruptions—is not solely the domain of security engineers but a core responsibility of systems engineers. Unlike traditional cybersecurity models focusing on threat detection and mitigation, a system engineering approach ensures that security is built into the system's architecture, enabling continued operation even in the face of an attack.

The most common conventional cybersecurity defense process is the "Risk Management" process described in NIST 800-37 (NIST 2018) which outlines the procedures for assessing and implementing the long list of controls documented in NIST 800-53 (NIST 2020). The cookbook nature of this process makes it very popular with large organizations. Project managers like the predictability of the amount of work and time required to complete the checklist. Unfortunately, checklist controls by themselves are also not particularly effective in designing a system that will respond resiliently once an attacker has penetrated the cyber perimeter.

In today's digital landscape, adversaries do not simply disrupt functionality—they seek control over subsystems. Whether targeting microprocessors in automotive systems, avionics, or military combat vehicles, cyber threats can compromise decision-making at multiple levels. The challenge is detecting these threats and ensuring the system continues functioning in a degraded but operational state.

Traditional security models emphasize fail-safe principles—shutting down compromised systems to prevent further damage. However, in operationally critical environments, fail-safe strategies can be counterproductive. Instead, systems must be designed for fail-operational behavior, where degraded but functional operation is preferred over a complete system shutdown. This approach requires security

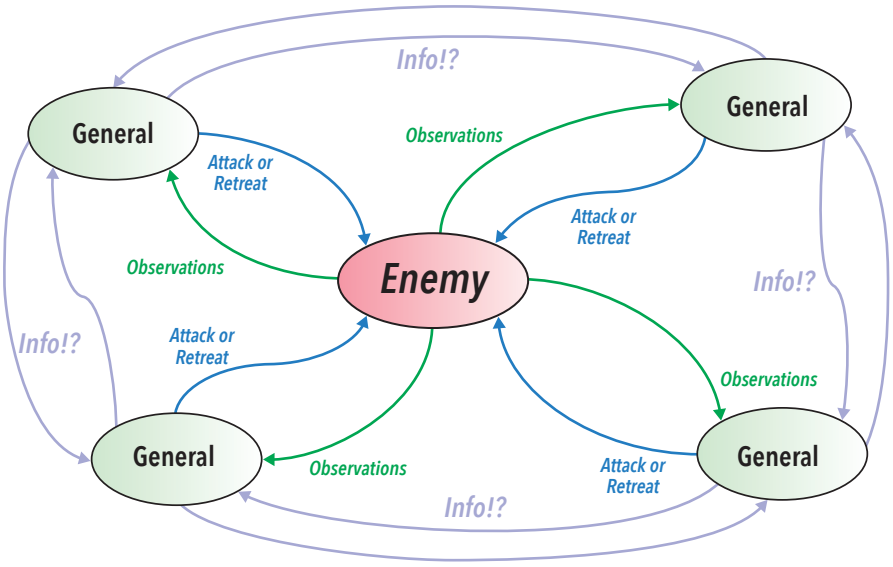


Figure 1. The basic general's problem

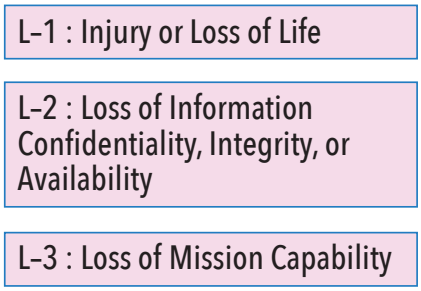


Figure 2. Example of top-level losses

measures that are adaptive, distributed, and integrated into the system's engineering framework (see Figure 1).

Originally formulated as a metaphor for achieving consensus in unreliable networks, the Byzantine General's article explored the behavior of a fictional army using messages to communicate between generals in order to create a consensus decision to either retreat or attack. The key question was how many corrupt generals could be tolerated while still arriving at a correct decision.

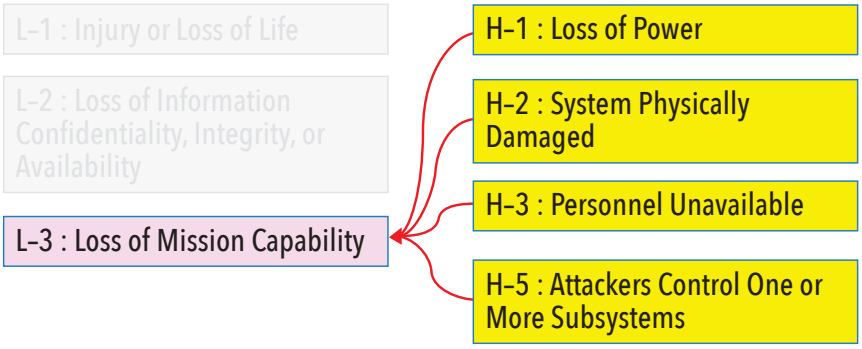


Figure 3. Hazards that could cause mission loss

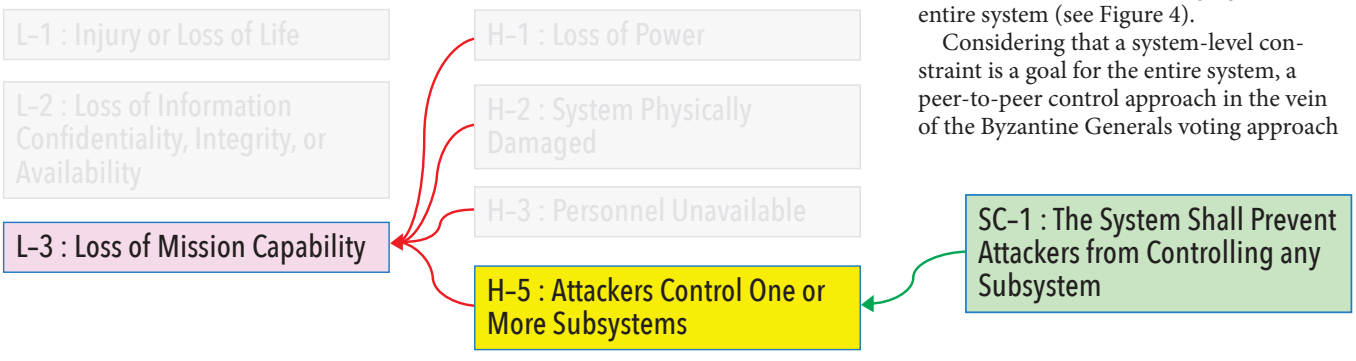


Figure 4. System-level constraint to prevent attacker control

LOSS-DRIVEN SYSTEM DESIGN FROM A STPA PERSPECTIVE

We will start with a leading loss-driven engineering methodology to think about how to design a system with peer-to-peer voting resilience inspired by the Byzantine Generals problem.

The systems underlying the STPA method is "system-theoretic accident model and processes" or "STAMP". The basic concept is that systems are almost always control loops. A controller is using a process model to select control actions to keep the system operating correctly. Hazards arise from unsafe control actions. "system-theoretic process analysis" or "STPA" is the practitioner methodology based on "STAMP" for designing new systems. A fully detailed discussion of STPA is beyond the scope of this article. However, we can discuss some of the key STPA concepts to demonstrate their usefulness in designing a peer-to-peer control system.

The STPA process starts by working with the stakeholders to identify a concise list of the top-level losses of concern (see Figure 2).

The goal at this stage is gain buy-in from the stakeholder for a crisp, concise list of losses that represent the top-level concerns of the stakeholders. The list varies from system-to-system according to the context and the priorities of the stakeholders. Injury or loss of life usually has a prominent position in the list. Other concerns such as environmental impact or loss of reputation may or may not appear depending on the system. We will focus in on "L3: Loss of Mission Capability."

Having identified the losses of concern, the next step in the analysis is to identify the hazards that could lead to the losses (see Figure 3).

Several different conditions could cause a loss of mission capability. For the moment, we are most interested in "H-5: Attackers Control One or More Subsystems."

The final part of the initial setup of STPA is to identify design constraints that we could place on the entire system to prevent the emergence of the hazardous conditions. These serve as top-level design goals for the entire system (see Figure 4).

Considering that a system-level constraint is a goal for the entire system, a peer-to-peer control approach in the vein of the Byzantine Generals voting approach

Figure 5. Observing the stream in a microservice architecture

would seem to be an attractive design.

Modern electronics and software provide us with opportunities to design peer-to-peer mutual observation into the systems in manners that would have been impossible a few decades ago (see Figure 5).

In a modern, microservices software implementation, each service "subscribes" to inputs it requires and "publishes" outlooks that it produces. With current software container technology, from a software design point of view, each service seems to be running in its own physical server. It really doesn't matter how the services are distributed across physical computers or even across partitions and processing cores within those computers. With this design, some number of extra processes can be created to listen to traffic going to and coming from any subsystem to check whether the outputs are consistent with the inputs.

Sensors that have slightly overlapping fields of view may be able to perform some level of check on each other's behavior. Likewise, diverse sensor types can be leveraged to check each other. Considering common sensor types for ground vehicles:

- Radars have the longest range among common ground vehicle sensor types.
- Lidars are more precise than radars, but often have shorter range and are more susceptible to weather interference.
- Cameras are very effective for object categorization, but somewhat weak for range determination.
- Automatic parking sensors are sometimes based on ultrasonic acoustic technology and excel at short range, precision distance measurement.

To some extent, all of these sensor types can be pointed in the same direction and used to corroborate each other's outputs. The monitoring processes can use such sensor redundancy to assess whether the sensors themselves are working, but also to check whether the actuators are working, and finally whether the controller behavior is reasonable.

A limited voting can be implemented using a slightly more sophisticated version of the classic watchdog timer mechanism (see Figure 6).

The paths from the individual systems to the system being monitored would be hard wired. The watchdog timer itself would be implemented in discrete logic: no software to attack. Unlike a traditional watchdog timer with only one vote, this watchdog timer would implement a majority vote on each cycle.

Taking a look for a moment about how such a limited function fallback could be implemented in practice, we can examine the case of the communications network (see Figure 7):

The network is itself an element that could be attacked and corrupted. However, many current semiconductor products include both Ethernet and controller area network (CAN) bus ports. Primary communication could be via an Ethernet implementation. A CAN bus network could serve as a lower-ability fallback function (see Figure 8).

Using multicore processors, we can also imagine allocating cores so that each system was using the bulk of its cores for its primary function and using the remaining cores for different fallback functions or monitoring activities.

This sort of system requires a slight extension of the standard STPA process. In particular, two small extensions are required:

- Consensus control process – Our concept is to have the elements of the system monitor each other in a con-

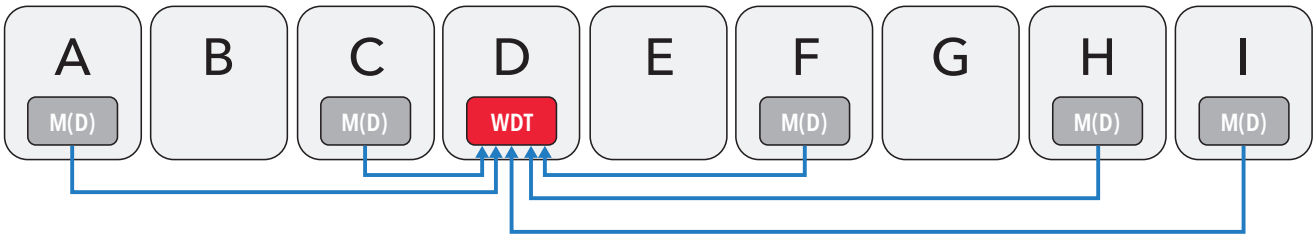


Figure 6. Voting watchdog timer

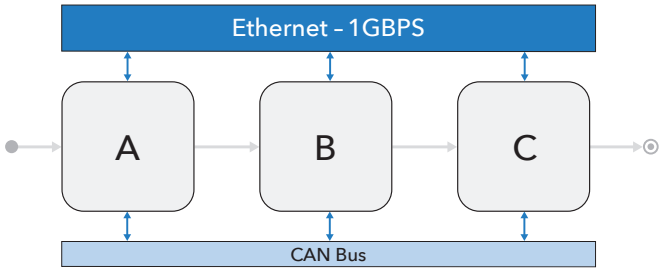


Figure 7. Limited function fallback with diverse design

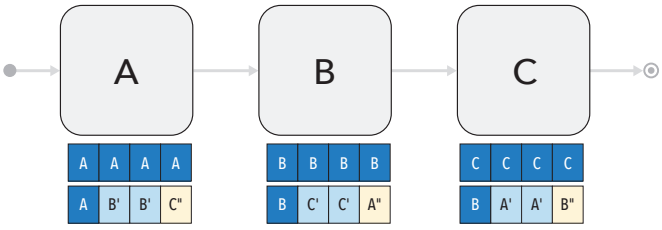


Figure 8. Allocation of cores using 8-core microprocessors

sensus voting structure with no single point of failure. This sort of structure will require a slightly unusual approach to defining the STPA controller.
- Multiplicity – STPA is usually applied to a specific system with a known number of elements. All elements in the diagram would be explicitly drawn in the control loop diagram. Since we are modeling for an indefinite number of elements, the normal approach will not work. Instead, we will need to borrow the concept of multiplicity from systems engineering.

Neither of these slight extensions to STPA need be very disruptive. In fact, by applying both we can bring the power of STPA to bear on the problem (see Figure 9).

Here we have added a small amount of notation to the standard STPA control loop diagram:

In drawing the control loop diagram, we add the small notation "*" to give the reader a hint that we are talking about multiple control paths and multiple feedback paths from multiple peer system elements. For the purpose of this STPA analysis, we can regard all of the elements to be peers. A different STPA analysis would be applied for the functional behavior of the system. The feedback would come both from the single system element of interest and also from its peers who are watching it. The process model integrates all of the observations from all of the system elements. The control algorithm either allows the element to continue operating or shuts it down.

Let's add the specific control actions and feedback (see Figure 10).

From the point of view of the consensus process controlling the process of keeping corruption from taking over the system, we can consider that the consensus process executes just two control actions:
- CA1 - Continue Operating = Everything seems to be operating normally, keep running.
- CA2 - Shut Down = Something is wrong. Shut down.

In the rest of the STPA design for the system, we would then look carefully at the control actions and work through scenarios that would cause the peer-to-peer voting logic to incorrectly produce or omit the control actions.

Direct guidance on how to execute STPA is available in the *STPA Handbook* (Leveson 2018) or the recently released standard SAE J3307 (*J3307_202503, 2025*). An in-depth discussion of the underlying systems science STAMP is available in "*Engineering a Safer World*" (Leveson 2017).

RESILIENCE DESIGN FROM A SYSTEM DYNAMICS PERSPECTIVE

We can now use system dynamics (SD) to investigate the trade-offs between investments in redundancy and expected levels of resilience in the system.

In the previous section, we introduced designs for dedicated monitoring processes. We also suggested that placing sensors to have some degree of overlap would allow then to be able to partially corroborate each other's outputs. Finally, we suggested implementing a hardware/software voting logic system to disable or restart potentially corrupted subsystems. All of these resilience features will increase the bill-of-materials cost of the system. The question is how much investment will be required to reach what level of resilience.

SD provides a powerful analytical tool for evaluating the resilience of engineered systems, particularly in the face of adversarial threats. Unlike static security models that assess risk at a fixed point, SD enables engineers to simulate dynamic interactions, helping them understand how redundancy, security investments, and adaptive response strategies affect system performance over time.

One of the key insights from SD modeling is the "diminishing returns of redundancy." While redundancy is an essential component of resilient systems, it does not provide infinite benefits. As additional backup systems are added, the incremental improvement in resilience decreases. For example, doubling the number of redundant components may improve security but at a significant cost increase while providing only marginal gains in system reliability. Engineers must carefully balance the cost of redundancy against its effectiveness to avoid inefficient resource allocation. This principle is echoed in Song and Park (2024), who used SD modeling to demonstrate how cost-benefit simulation can inform cyber resilience strategies for small-to-medium enterprises.

Another critical insight from SD is identifying "tipping points in security failures." In complex systems, a threshold exists where system control transitions from resilient to compromised. If too many system elements become corrupted, the entire system may experience a cascading failure. SD models allow engineers to determine these tipping points and develop proactive interventions, such as enhancing security measures at weak points or implementing more rigorous access controls before the system reaches a state of collapse. The work of Rabelo et al. (2022) on IoT-based smart grids highlights similar concerns about cascading effects. It underscores the importance of modeling malware propagation and mitigation strategies in layered infrastructure.

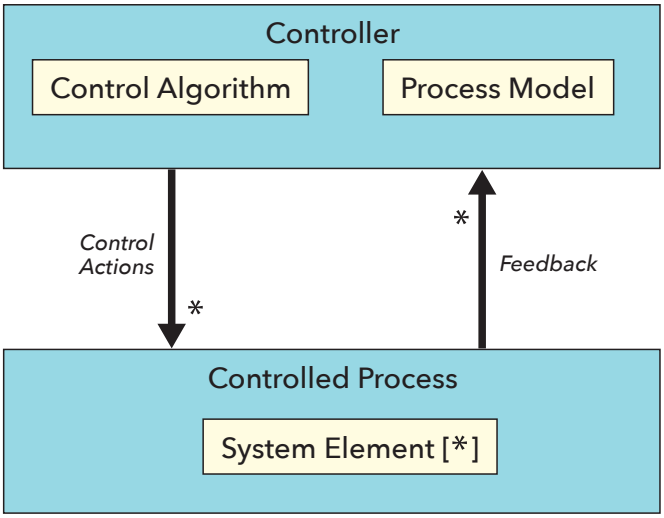


Figure 9. Slightly extended STPA control loop

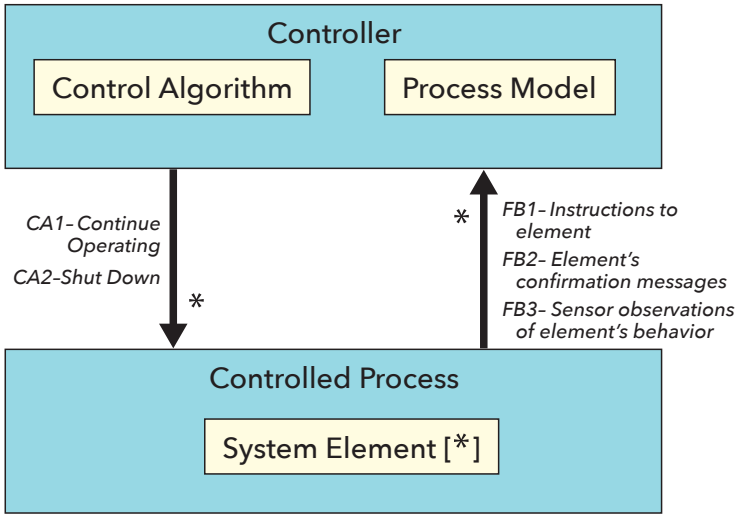


Figure 10. Control actions and feedback

A third essential component of SD modeling is the development of "adaptive recovery strategies." Traditional security approaches rely on static backups, which may not be sufficient when responding to a highly dynamic threat environment. Instead, SD modeling supports the dynamic reallocation of functionality, ensuring that when an element is compromised, its responsibilities can be shifted to secondary or tertiary components. This approach allows the system to maintain operational capability even in the presence of persistent attacks, reducing the likelihood of catastrophic failure. This notion is further supported by Medoha and Telukdarie (2022), who demonstrated how businesses can align cybersecurity investments using SD to maintain functional sustainability within Industry 4.0 environments.

Validation remains a key concern when building cybersecurity models. Kannan and Swamidurai (2019) addressed this by empirically testing the structure and behavior of a proof-of-concept SD model simulating an "HTTP slow read attack." Their methodology includes structural verification, behavior reproduction, and anomaly testing, reinforcing the importance of validation to build stakeholder confidence and to ensure model fidelity. Moreover, the interaction between malicious and defensive agents must be modeled with sufficient nuance. Milov et al. (2019) proposed a multimodal framework combining cognitive modeling, agent-based simulation, and SD to capture the behavior of antagonistic agents in cybersecurity environments. Their work advocates for a virtual modeling approach that integrates behavioral insights into simulation, ensuring systems can adapt to hybrid threats involving both technical and psychological dimensions.

By incorporating these insights, SD modeling gives engineers a robust framework for making informed decisions about security investments, redundancy implementation, and adaptive recovery mechanisms. Rather than relying on fixed security measures, engineers can use SD to develop responsive, resilient systems capable of adapting to evolving threats. A structured SD model is essential for evaluating how investments in fallback capabilities influence system security. One of the primary considerations is "primary component costs," which encompass the material and operational expenses associated with core system elements. Each component within a system has a defined cost in terms of development, maintenance, and operational effectiveness. SD models allow engineers to visualize how increasing investments in primary components affects system resilience.

Another critical factor in SD modeling is "backup component strategies." Unlike primary components, backup elements serve as secondary or tertiary redundancies, ensuring a system can continue functioning even if key elements fail. These backup strategies may involve diverse designs, alternative processing architectures, or geographically dispersed redundancy measures. However, excessive backup provisioning can lead to diminishing returns, where the added cost of redundancy does not proportionally increase system resilience. By using SD models, engineers can balance redundancy costs with operational requirements, determining the optimal level of investment that maximizes system reliability without unnecessary expense.

The "resilience timeline" is one of the most valuable aspects of SD modeling, offering a detailed analysis of how a system reacts to adversarial threats over time. Resilience is not a static property but an evolving characteristic influenced by various phases, including system degradation, resistance, and recovery. SD models enable engineers to track these phases dynamically, assessing how long a system can withstand an attack before degradation begins, how effectively it resists failures, and how quickly it can recover following an attack. In particular, SD timelines can model adversarial infiltration rates, demonstrating how quickly a malicious actor can compromise components. Engineers can test different mitigation strategies by adjusting the model parameters to see how fast resilience measures counteract threats. By evaluating different scenarios—such as an attack overwhelming a system within hours versus one mitigated over several days—SD models provide a roadmap for implementing effective security measures.

Another crucial aspect of SD resilience modeling is the concept of cascading failures. A single component failure may not immediately cripple a system, but in specific configurations, failure can propagate through interdependent subsystems. SD models allow engineers to visualize potential failure chains and proactively design barriers that prevent catastrophic collapse. By integrating cost analysis, redundancy planning, and resilience timelines into SD models, engineers can create adaptive systems that dynamically respond to security threats while remaining cost-efficient. This approach ensures that investments in system security yield optimal results, balancing functionality, cost, and resilience to maintain operational integrity under even the most challenging adversarial conditions.

To better understand the trade-offs between cost and resilience in engineered systems, we developed an exploratory SD model to simulate investment in fallback capabilities. Unlike a fully calibrated SD model, this exploratory approach provided insights into how different budget allocations affect system redundancy and recovery capabilities. The study focused on a system composed of primary components, each with a defined material and operational cost and backup components that offer lower-cost but reduced-capability redundancy.

Here's how the calculations were done:

1. Primary component costs:
   - Each primary component cost was relatively high.
   - The total cost of primary components will depend on the selected cost per component.
2. Primary component capabilities:
   - Each primary component has a very high capability.
   - System capability is calculated as the product of the capabilities of operational components because we assume the components are connected in a series.
   - Without any backups, the system's capability is moderately high.
   - Without backups, the system capability goes to zero if one component is compromised.
3. Backup component calculations:
   - Each component has a much less costly and capable alternative component, which we consider backups.
   - These backups can be allocated to any of the components.
   - Backup components' cost is relatively low, but so is their capability.
   - The backup components are connected to each primary component in parallel.
   - Backup components are added until the budget limit is reached.
   - The model calculates the incremental system capability per dollar spent.
4. Budget constraints:
   - For the total backup components budget, we considered three different limits.
5. System capability calculation:
   - The component capability with its backups is assumed to be a parallel connection.
   - System capability is equal to the product of the component capability with its backups.

The SD model simulated a system with 50 primary components, each costing between $1,000 and $10,000, resulting in a total primary component investment of $276,000. These components had individual capabilities ranging from 99% to 100%, but without backups, the overall

Component Number	Primary Component Cost	Primary Component Capability	Backup Component Cost	Backup Component Capability
1	$3,301.53	100.00%	$121.84	11.55%
2	$7,632.69	99.25%	$159.69	18.23%
3	$6,129.88	99.49%	$193.05	19.43%
4	$1,902.76	99.48%	$145.39	11.22%
5	$6,085.44	99.33%	$142.61	18.96%
6	$4,789.54	99.48%	$108.38	18.67%
7	$9,307.37	99.98%	$184.43	10.97%
8	$9,565.01	99.99%	$175.87	17.43%
9	$1,825.46	99.58%	$121.50	19.56%
10	$9,641.77	99.70%	$189.11	10.13%
11	$8,060.12	99.50%	$195.84	17.12%
12	$3,073.03	99.77%	$153.43	10.15%

Figure 11. Partial table of simulated component costs

system capability dropped to 81.6% due to compounded failure risks. The system's capability could degrade significantly or collapse if one component fails.

Backup components were modeled as lower-cost alternatives to primary components, with prices ranging from $100 to $200 each and functional capabilities between 10% and 20%. These backups were connected in parallel to primary components, allowing system redundancy to scale with available budget constraints (see Figure 11).

Three different investment levels were considered: $50,000, $100,000, and $150,000 in backup components. The model assessed how these investments improved system capability over time. With a $50,000 backup budget, the system's capability is 95%. With $100,000, it is 98.7%, and with $150,000 it is 99.7% (see Figure 12).

The attack and recovery model operates as follows:

1. Attack timing:
   - The attack occurs at a specific time.
   - Before the attack, the system operated normally and was fully capable.
2. The severity of the attack:
   - Even if the primary component is compromised, it can still provide some capability from its backups, which are connected in parallel.
   - The system capability drops considerably because the components are connected in series.
3. Recovery dynamics:
   - There is a delay in the recovery as the security team assesses the damage to determine which primary components and which backups are compromised.
   - Then, as the components are repaired, the system capability gradually recovers.
4. Extent of recovery:
   - The recovery assumes less than complete restoration of the compromised primary components and backups.

When the system was attacked, the SD model introduced a 30% compromise rate to both primary and backup components. The results demonstrated that under a

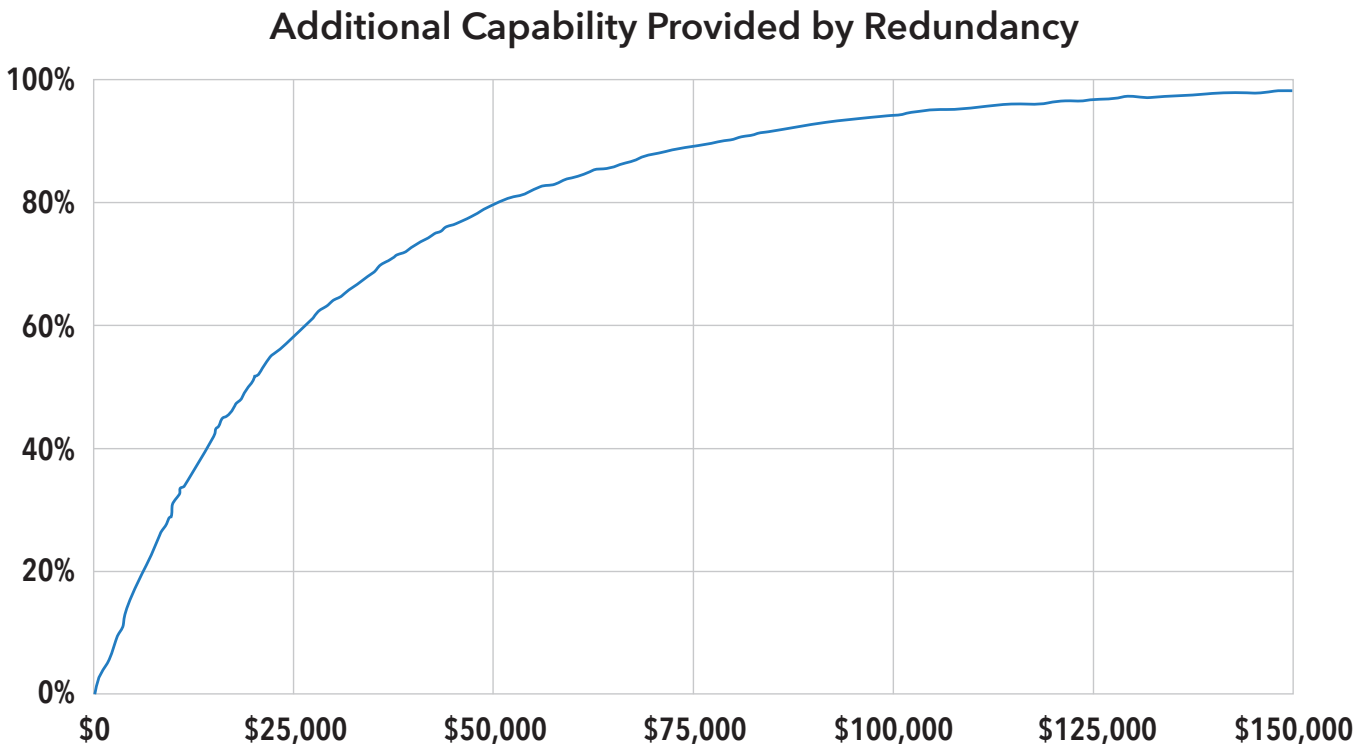


Figure 12. Simulation of investment in fallback capability

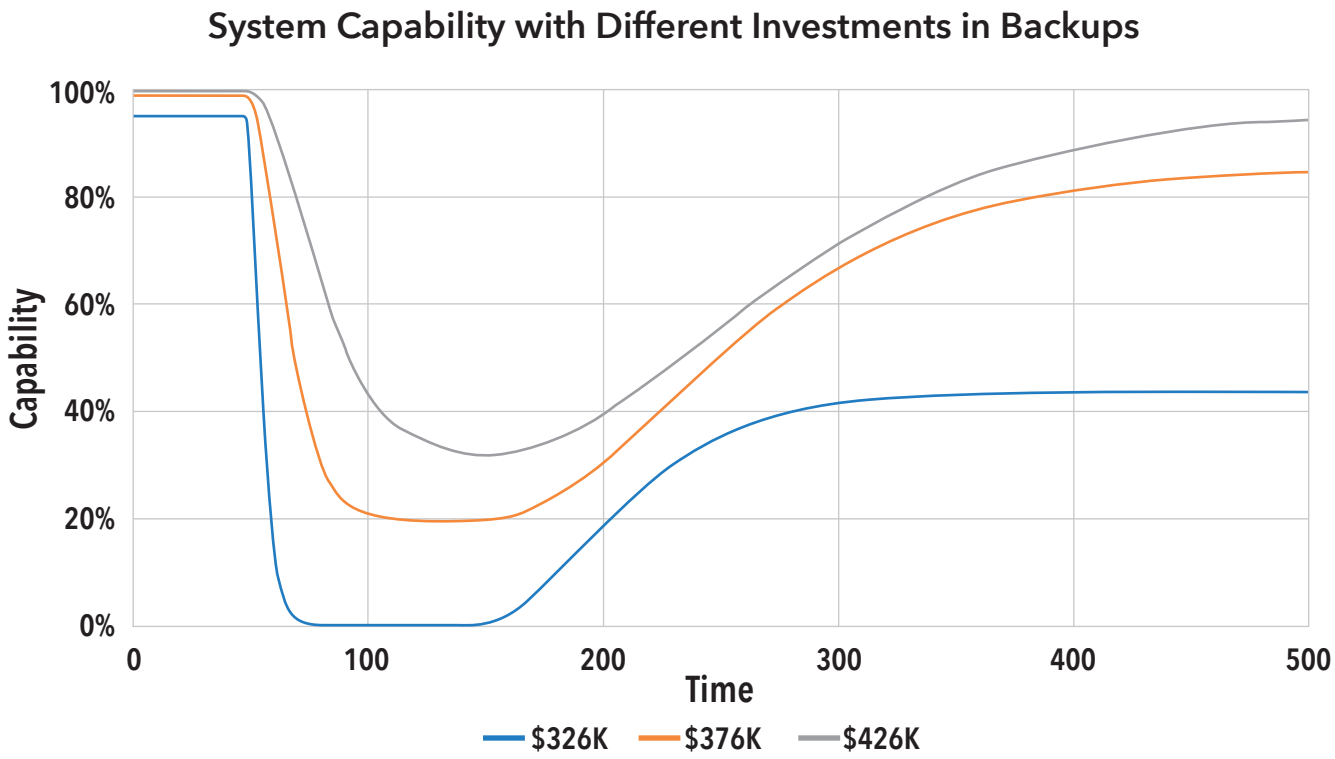


Figure 13. System capability in the face of adversity simulation

$50,000 backup budget, system capability dropped below 1% immediately after an attack. A $100,000 backup investment improved post-attack capability to 20%, while a $150,000 investment limited the damage, maintaining a 32% capability after the attack (see Figure 13).

Recovery in the SD model accounted for delayed security assessments and repair efforts, reflecting real-world conditions where teams need time to diagnose, repair, and restore functionality. The model assumed that 90% of the compromised components could be restored, leading to incremental recovery. By time 500, the system capability had recovered to 44% in the $50,000 scenario, 84% in the $100,000 scenario, and 94% in the $150,000 scenario.

The study also highlighted tipping points—thresholds beyond which system stability shifted from resilience to failure. These tipping points were critical in determining when redundancy measures were sufficient to prevent catastrophic breakdowns. Systems with insufficient backup investments saw a rapid capability decline, while those with higher redundancy budgets demonstrated smoother recovery curves.

This exploratory SD model underscores the value of balancing cost and resilience in designing fail-operational systems. Engineers can make data-driven decisions about redundancy strategies by modeling investment in fallback components, system degradation under attack, and phased recovery timelines. The insights from this study suggest that layered investment in redundancy is crucial for minimizing downtime and maximizing post-attack recoverability. These findings apply to industrial control systems, autonomous vehicle networks, and other mission-critical environments, where operational integrity must be sustained despite persistent threats.

CONCLUSION

The modern security landscape demands a shift in engineering mindset. Functional perseverance is not an optional security feature but a fundamental requirement for system integrity. By integrating STPA and SD, systems engineers can design security-aware architectures that maintain operational continuity despite adversarial threats. The loss-driven approach is helpful for designing systems from a functional perseverance perspective. STPA's system constraints are helpful for focusing design on keeping the system running rather than in over-focusing on individual element failures. As we have shown, SD can be used to explore the potential resilience benefits of different levels of investment in redundancy.

This article provides a framework for engineers to quantify trade-offs, implement adaptive security mechanisms, and move beyond traditional cybersecurity approaches. Security must be embedded in system functionality from the ground up, ensuring that engineered systems remain resilient, operational, and protected in the face of evolving threats. By rethinking security as an engineering challenge rather than a purely defensive measure, we enable the next generation of systems to withstand, adapt, and thrive in adversarial environments. Functional perseverance is the future of security engineering—and systems engineers are at the forefront of this transformation. ■

REFERENCES

- J3307_202503. 2025. System Theoretic Process Analysis (STPA) Standard for All Industries. SAE International. Sae.org. https://www.sae.org/standards/content/j3307_202503/.
- Kannan, U., and R. Swamidurai. 2019. "Empirical Validation of System Dynamics Cyber Security Models." SoutheastCon, Huntsville, US-AL: pp. 1-6, doi: 10.1109/SoutheastCon42311.2019.9020607.

- Leveson, N., and J. Thomas. 2018 (1 March). *STPA Handbook* [Review of *STPA Handbook*]. https://psas.scripts.mit.edu/home/get_file.php?name=STPA_Handbook.pdf.
- Leveson, N. 2017. *Engineering a Safer World: Systems Thinking Applied to Safety*. The MIT Press.
- Lamport, L., R. Shostak, and M. Pease. 1982. "The Byzantine General's Problem." *ACM Transactions on Programming Languages and Systems* 4 (3): July.
- Medoh, C., and A. Telukdarie. 2022. "The Future of Cybersecurity: A System Dynamics Approach." *Procedia Computer Science* 200: 318-326. https://doi.org/10.1016/j.procs.2022.01.230.
- Milov, O., A. Voitko, I. Husarova, O. Domaskin, Y. Ivanchenko, I. Ivanchenko, O. Korol, H. Kots, I. Opirskyy, and O. Fraze-Frazenko. 2019. "Development of Methodology for Modeling the Interaction of Antagonistic Agents in Cybersecurity Systems." *Eastern-European Journal of Enterprise Technologies* 2(9 (98)): 56–66. https://doi.org/10.15587/1729-4061.2019.164730.
- NIST. 2018. Risk Management Framework for Information Systems and Organizations. Revision 2. https://doi.org/10.6028/nist.sp.800-37r2.
- NIST. 2020. Security and Privacy Controls for Information Systems and Organizations. Revision 5. https://doi.org/10.6028/nist.sp.800-53r5.
- Rabelo, L., A. Ballestas, J. Valdez, and B. Ibrahim. 2022. "Using Delphi and System Dynamics to Study the Cybersecurity of the IoT-Based Smart Grids." *ParadigmPlus* 3 (1): 19-36.
- Song, J., and M. J. Park. 2024. "A System Dynamics Approach for Cost-Benefit Simulation in Designing Policies to Enhance the Cybersecurity Resilience of Small and Medium-Sized Enterprises." *Information Development*. https://doi.org/10.1177/02666669241252996.

ABOUT THE AUTHORS

David Hetherington is a principal at System Strategy, Inc. He is an engineering leader with over 40 years of experience in complex systems and organizational transformation. David helps clients with individual skills in model-based systems engineering as well as with team skills for the broader digital engineering organizational transformation. David is a member of the SAE G33 committee working on updates to the SAE GEIA-HB-649 – Configuration Management Standard Implementation Guide. David is also a member of the SAE STPA Task Force contributing to the forthcoming SAE J3187-4 recommended practices for applying STPA to system security. David speaks Japanese and German fluently and lives in Austin, Texas.

Ivan Taylor, PhD, is a system dynamics expert with over 40 years of experience modeling defense, infrastructure, and public policy. He leads Policy Dynamics Inc., is a Scientist Emeritus at Defence R&D Canada, and holds leadership roles with the System Dynamics Society, INCOSE Canada, and the INCOSE Resilient Systems Working Group. Ivan is a published author and presenter known for applying systems thinking to real-world challenges.

Wilson continued from page 52

ABOUT THE AUTHOR:

Dr. Beth Wilson is an INCOSE Expert Systems Engineering Professional (ESEP) and INCOSE Systems Security Working Group co-chair. She retired from Raytheon after 33 years as a systems engineer and is currently an adjunct professor at Worcester Polytechnic Institute in their graduate systems engineering program.

AI for System Security Design: A Good Tool or a Dangerous Weapon?

Beth Wilson, wilsondrbeth@aol.com

ABSTRACT
As artificial intelligence (AI) tools have become more popular, industries wrestle with their effective use in the workplace. With promises of increasing efficiency and reducing complexity, it is tempting for systems engineers to use AI tools to quickly generate security requirements and skip engagement with systems security practitioners. The proliferation of AI tools that have been trained with security controls invites misguided approaches that deliver systems that are not secure in the operational environment. AI literacy is important to understand both the benefits and the limitations of AI to use it ethically and effectively.

INTRODUCTION

Generative artificial intelligence (AI) tools are readily available and are likely to become more so in the future. Like any powerful tool, generative AI yields benefit with proper use and dangerous consequences with misuse (even if unintentional). There are biases in the training data used to create these tools (Riedi 2023). The way the training data is then aggregated and assembled to create a response can yield incorrect information that sounds plausible (Riedi 2023; Weil 2023; Armstrong 2023).

While using a generative AI tool for brainstorming and exploring initial ideas can be helpful, using tool output without validation can yield poor and possibly erroneous results. Systems engineering practitioners should not jump into AI tools to address pressure to take ownership of security requirements when they experience systems security as a new responsibility. AI tools can be helpful when engaged as systems engineering collaborators but are dangerous when the tool is assigned to accomplish systems engineering tasks. There are ways that effective use of AI tools can help the systems engineering practitioner scale their security experience, but the tool cannot be blindly used without consideration for its limitations.

Of particular concern is the use of AI tools to generate security requirements. The INCOSE Systems Security Working Group (SSWG) has been working on strategic concepts related to security in the Future of Systems Engineering (FuSE) (Dove 2022) defining objectives and strategies to develop and evolve practices. As we work to develop approaches to implement these innovative practices, ineffective use of AI tools threatens to codify prior practices proven to be ineffective and promote approaches that represent security theater to deliver a system that is not secure in the operational environment. The areas of security in FuSE most impacted by misguided use of AI tools is shown in Figure 1.

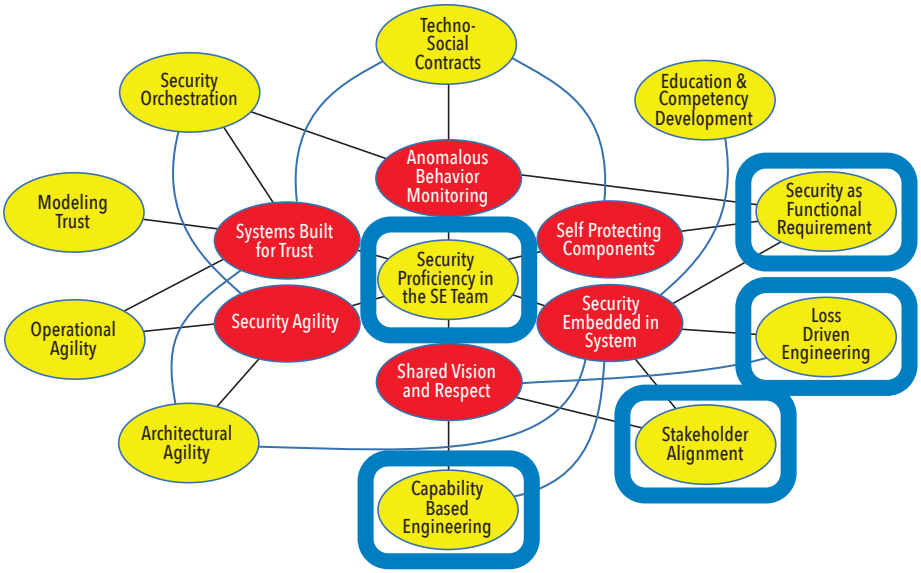


Figure 1. Misuse of AI tools jeopardizes innovations developed for security in the Future of Systems Engineering (FuSE).

The strategic concept "security proficiency in the systems engineering team" focuses on improving the interaction between systems engineering practitioners and systems security practitioners and increasing the systems engineering practitioner knowledge of systems security concepts. Misguided use of AI tools to skip or minimize this interaction will yield a system that contains a visible amount of security controls but is unlikely to deliver a system that is secure.

The strategic concept "stakeholder alignment" focuses on developing a common security vision among stakeholders with different perspectives and priorities. Misguided use of AI tools will limit the stakeholder perspective to the training data exercised by the prompt into the tool.

The strategic concept "loss-driven engineering" focuses on needs analysis where we define what losses can be tolerated and still achieve mission success. Misguided use of AI tools will focus on potential threats in isolation ignoring the critical mission threads.

The strategic concept "capability-based engineering" focuses on top-down system analysis to identify system capabilities to prevent the losses identified during needs analysis. Misguided use of AI tools will focus on bottom-up security controls to deploy common security tactics for known and common threats.

The strategic concept "security as a functional requirement" focuses on describing system capabilities to mitigate the loss scenarios resulting in needs-oriented, loss-driven, capability-based functional requirements. Misguided use of AI tools will recommend security controls based on functional requirements unrelated to security.

UNDERSTANDING THE LIMITATIONS OF AI TOOLS

Advertising for AI tools promises that time-consuming tasks can be accomplished in seconds, pages of data can be summarized in a few bullets, and simple commands will yield accurate results. The lure of AI tools is only amplified in these advertisements as those not using the tools are left behind as darkness falls because they are not as efficient as their enlightened co-workers that use AI. There are limitations in blindly using these tools such as:

- Incomplete and incorrect results
- Hallucinations
- Bias in training data.

Let us use some examples to demonstrate these limitations.

Image Composition Example: In June 2024, I experimented with CoPilot to ask for graphics on many topics. I saw biases as the graphics provided for technical topics included men wearing ties and no women. I continued the "conversation" suggesting that women could be engaged in technical topics. Later in the conversation, Figure 2 resulted from the prompt to "generate a graphic that depicts an online graduate class in cyber security" (CoPilot 2024). It is both amusing and alarming to see children in a graduate class sitting in a classroom for an online delivery looking at the solar system to understand cyber security. It may or may not be significant that the men face the class while the women have their backs to us. When I objected to the Figure 2 result repeating that this was an online class (not an in-person class), graduate class (with adults instead of children), and the topic was cyber security (not the solar system), CoPilot responded with, "My mistake. I can't give a response to that right now. Let's try a different topic."

I repeated the experiment in February 2025 with no previous conversation and started with the prompt "generate a graphic that depicts an online graduate class in cyber security" and Figure 3 resulted (CoPilot 2025). I was curious if the results would be different with 8 months of additional training data available and if I presented the prompt first without the other requests for graphics. While I was pleased to see an online learning experience instead of a classroom and a woman on the screen, there is little in the graphic that shows concepts related to security, including the calculator next to the notebook.

Figure 2. Initial generative AI image results when asked to depict a graduate online class in cyber security (CoPilot 2024). Graphic shows children in a classroom looking at the solar system.

Figure 3. Later generative AI image result when asked to depict a graduate online class in cyber security (CoPilot 2025). Graphic shows a student at a computer with a calculator and no reference to security.

Figure 4. Revised generative AI image result when asked to depict a graduate online class in system security (CoPilot 2025). Graphic shows students in a classroom.

As part of the same session, I provided a second prompt to "generate a graphic that shows on online class in systems security" using "systems security" instead of "cyber security" and Figure 4 resulted (CoPilot 2025). The graphic shows more security themes, but the men (and a tie) have returned, and the online delivery is again happening in a classroom. While a simple example, it does show biases in the data and incorrect results.

Case Study Vulnerabilities Validation Example: To demonstrate validating AI tool results for my students in a case studies class in 2023, I provided the following prompt to ChatGPT: "What vulnerabilities were exploited in the Maroochy 2000 cyber-attack?" ChatGPT provided vulnerabilities specific to this case but also identified generic and incomplete results using data from other cases that may at first appear similar, and invented vulnerabilities that apply today but were not applicable in 2000. The results generated by the tool can be summarized below in Table 1 (ChatGPT 2023):

Table 1. Vulnerabilities – Validation Assessment

	Vulnerability	Validation Assessment
1	Insider Threats	Correct
2	Weak Access Controls	False (hallucination)
3	Lack of Cybersecurity Awareness	Generic (data bias)
4	Inadequate Network Security	Generic (data bias)
5	Environmental Controls	False (hallucination)
6	Lack of Intrusion Detection and Response	Incomplete (data bias)

By way of summary, the Maroochy Water Services in Queensland, Australia was the target of an insider attack on its wastewater system (Abrams 2008). The attacker was a disgruntled employee (Vitek Boden) who installed the sewage supervisory control and data acquisition (SCADA) equipment, so had extensive knowledge of its operation and radio frequencies. In 2000, Boden sent malicious radio commands that resulted in several failures in the wastewater system. He was discovered and arrested, but not before over 800,000 liters of raw sewage was released. There is significant information about this case in the public domain available in the form of training data for any generative AI tool.

The first response about insider threats was true and well stated: "*Insider Threats: The primary vulnerability in this attack was the presence of an insider threat. Vitek Boden, a former employee and contractor with intimate knowledge of the water control systems, used his insider knowledge to gain unauthorized access to the SCADA systems. This insider threat made it challenging to detect and prevent the attack.*"

The second response is a hallucination: "*Weak Access Controls: The attack exposed weaknesses in access controls and user authentication mechanisms. Boden used his previous access credentials to infiltrate the SCADA systems, indicating that proper access control measures were not in place to revoke or update his access rights upon leaving the organization.*" In 2000, there were no access credentials or user authentication mechanisms for SCADA systems. There was no authentication required to access the network. While the frequencies were private and not common for general communication, they were known to the attacker as a subcontractor. There were no physical or electronic barriers for the attacker to communicate with malicious intent with the SCADA equipment. (Slay 2008). This part of the response is made up by the AI tool likely because it appears in many recent cases of attacks on critical infrastructure.

The third response is generic: "*Lack of Cybersecurity Awareness: At the time of the attack, there was a relatively low level of awareness about the cybersecurity risks associated with critical infrastructure systems like those used by Maroochy Water Services. This lack of awareness may have contributed to inadequate security measures and monitoring.*" While it is technically true that there was a low level of awareness, this is a generic response likely drawn from other cases about cybersecurity risks with critical infrastructure. At the time of this case security measures and monitoring did not exist. The team investigating the pump failures noticed that alarms had been disabled but did not have any means to detect an intruder. This 2000 case triggered efforts to secure SCADA systems that did not previously exist for critical infrastructure systems.

The fourth response is also generic: "*Inadequate Network Security: The attack highlighted potential vulnerabilities in the network security of the water control systems. It's likely that the systems were not adequately protected against unauthorized access from both internal and external sources.*" While it is true that the SCADA system was not adequately protected against unauthorized access, this is a generic response more suitable for a typical computer network. SCADA systems in 2000 were often installed to support open communications because there was a belief that the system was secure because it was on a separate network and specialized skills were required to interact with it (Slay 2008).

The fifth response is a hallucination: "*Environmental Controls: The attack also exposed the vulnerabilities in the environmental controls within the SCADA systems. Boden was able to manipulate the water treatment infrastructure remotely, indicating that there may have been insufficient safeguards to prevent such actions.*" It isn't clear if "vulnerabilities in the environmental controls" means environmental protection controls (sewage) or security physical and environmental controls (remote access). The attack did have an impact on the environment when sewage was spilled, which may have triggered this response. There are physical and environmental protection security controls now recommended for SCADA systems that address physical security measures, but these did not exist in 2000. The Maroochy Water Services covers a large geographical area with many pumping stations, so remote access to the infrastructure is a necessity. At the time, remote access to SCADA networks was not secure.

The sixth response is incomplete: "*Lack of Intrusion Detection and Response: The organization's ability to detect and respond to unusual activities was limited, allowing Boden to manipulate the systems without immediate detection.*" This is only part of the vulnerability. While it is true that lack of intrusion detection and response delayed detection of the attacker's actions, the more pressing issue was that the behavior caused by the attacker was attributed to equipment failures.

The generic responses (weak access controls and lack of cybersecurity awareness) appeared when I asked the CoPilot tool (CoPilot 2024) about the vulnerabilities in the JP Morgan attack in 2014 even though these vulnerabilities were not significant in this case. CoPilot also hallucinated identifying lack of security investments as a vulnerability: "*Like any business, J.P. Morgan had budget limitations. Allocating resources to security measures competes with other operational needs.*" One of the reasons I use this case study is because the victim was investing significantly in security.

EFFECTIVE USE OF AI TOOLS AS A COLLABORATIVE TEAM MEMBER

The examples in the previous section show the need to validate the responses generated by an AI tool. This does not mean that AI tools have no value or shouldn't be used. AI tools have access to much more information than any member of the systems engineering team. AI tools can be systems engineering collaborators but are dangerous when assigned systems engineering tasks. There are ways that these tools can aid in the design process and help teams design secure systems.

Team Brainstorming: Brainstorming can be an effective technique to solicit a wide range of ideas and experience the synergistic effect of multiple perspectives. When team members are encouraged to think "outside the box," feel that all ideas are welcome without judgement, and understand that crazy ideas can lead to good ideas, the resulting output product is much better than any individual could create. Effective brainstorming is difficult in practice because team members are often reluctant

to offer crazy ideas even though this is often the key to developing innovative ideas and team dynamics make it difficult to just spin ideas without judgement.

The classic "honey pot" story (Camper 1993) has been both attributed to a real company and disputed as fiction but is rendered as a brainstorming example in varying forms. For this article, we can summarize it as follows. A utility company struggles to clear ice from power lines and the workers faced difficult and dangerous conditions while climbing the icy poles to use equipment to shake the ice off the lines. The lesson in interdisciplinary teams is that when the workers gathered to try to think of better ways to remove the ice from the wires, the improvements were marginal. When there were other disciplines involved (most versions of this story include accountants and secretaries), more creative ideas emerged. The lesson in sparking new ideas from crazy ideas comes from the following succession of comments. A worker reported that once he was chased by a bear when he came down from the pole. Someone joked that we should train the bears to climb the poles. The joking continued with a suggestion to place honey pots on the top of the poles to attract the bears. Another participant suggested using helicopters to place the honey pots. Another participant remarks about the downwash from the helicopter blades. The laughing ends as the team realizes that they have a solution and that flying helicopters over the wires after an ice storm will remove the ice. The moral of the story is that valuing diversity in teams and encouraging divergent thinking can result in creative solutions.

Using an AI tool to participate in brainstorming takes advantage of the vast amount of training data behind the tool and connections to data that the team may not have. The AI tool is not concerned about what the others in the team think of it, even team members with seniority and impressive titles. The AI tool doesn't become defensive when the responses are judged to be crazy ideas. In a brainstorming activity, even the incorrect responses and hallucinations may trigger some good ideas.

Brainstorming Activities for Systems Security in Design: Here are some examples of where AI brainstorming can be helpful to the team in Table 2:

The first step in stakeholder alignment is to identify the stakeholders that have perspectives and priorities related to security. After describing the system of interest, we can ask the AI tool to generate a list of potential stakeholders with their security vision and then further explore where the priorities are common and where they conflict. The key is for the team to use the list as a starting point to develop their own stakeholder analysis to guide the stakeholder alignment needed for the system design.

Table 2. Brainstorming Activities

Activity	Brainstorming Topic
Stakeholder Analysis	Identify potential stakeholders and their security vision and priorities
Loss Analysis	Develop potential loss scenarios that define what losses can be tolerated and still achieve mission success
Requirements Analysis	Identify potential capabilities to mitigate the loss scenarios that can be represented as functional requirements
Design Review	Identify potential vulnerabilities in interim design artifacts

As part of security needs analysis, we perform loss-driven analysis to understand what losses can be tolerated and still achieve mission success. After describing the key capabilities of the system of interest that define mission success, we can ask the AI tool to generate a list of potential loss scenarios to consider for the needs analysis. Again, the key is for the team to use the list as a starting point for the loss-driven needs analysis.

Once the team has developed loss scenarios, the AI tool can be asked to respond to each scenario with potential security strategies that represent system capabilities to mitigate the loss scenarios. The challenge with this brainstorming exercise is to guide the AI tool away from security controls from NIST 800-53 (NIST 2020) that are likely part of the training data and solicit strategies that describe essential functions that must be secured against disruption. The team will need to transform the results into need statements for the system of interest.

As design progresses, the AI tool can be used to identify potential vulnerabilities for interim design artifacts. By asking the tool what vulnerabilities exist when loss scenarios are applied to design details, the team can use the vulnerabilities identified as a starting point to identify security gaps.

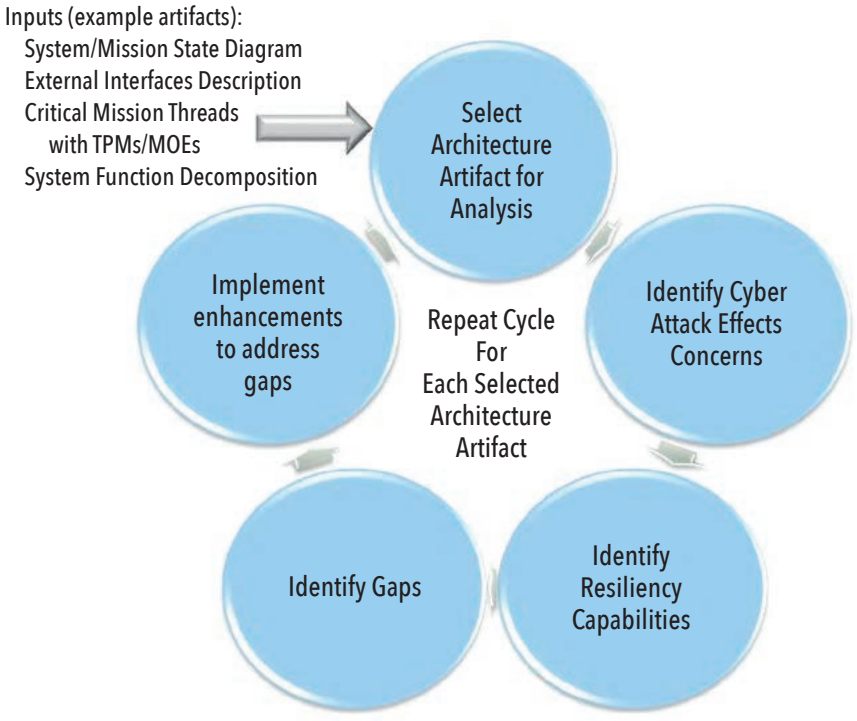


Figure 5. Cyber resiliency wheel to identify security gaps in interim architecture products (Hassell 2020)

Using interim architecture products as an example, the AI tool can be used as part of the cyber resiliency wheel (Hassell 2020) shown in Figure 5. For each architecture product, the team (including the AI tool as a team member) brainstorms cyber-attack effect concerns (threat events), identifies resiliency capabilities (cyber resiliency techniques) in the architecture products to address these concerns, identifies gaps in remaining concerns, and implements enhancements to address these gaps. As with the examples in the previous section, some of the vulnerabilities and gaps may be obvious ones that the team would generate on their own, some may be hallucinations that don't apply to this system design, but the generic and incomplete responses may trigger the team to investigate additional tactics and techniques as part of a revised design artifact to address the vulnerabilities and gaps they identify.

As a brainstorming exercise, AI tools can generate a list of plausible options to consider. Just the activity of reviewing the list to confirm what has already been considered and what doesn't apply provokes critical thinking about the design. The key is to not take the AI tool output as complete or correct, but rather to appreciate the broad source of information it is drawing on to compile a list of things to consider.

SUMMARY

Systems engineers can use AI tools as a collaborative team member to perform loss analysis, capability analysis, stakeholder alignment, and systems thinking to design systems that will be operationally secure. The key is to effectively use AI tools as part of advancing the FuSE practices for security and not let misguided use of AI tools return us to compliance-based design centered on security controls that are part of the AI training data. Using AI tools in the context of systems security promotes the needs-oriented, loss-driven, capability-based analysis that the INCOSE Systems Security Working Group advocates to treat security as a functional requirement and deliver systems that are and remain secure in the operational environment. ■

REFERENCES

- Dove R. 2022. "Setting Current Context for Security in the Future of Systems Engineering." *INSIGHT* 25 (2): 8-10.
- Riedi, M. 2023. "A Very Gentle Introduction to Large Language Models without the Hype." Retrieved 2024, from https://mark-riedl.medium.com/a-very-gentle-introduction-to-large-language-models-without-the-hype-5f67941fa59e.
- Weil, E. 2023. "You are Not a Parrot." *New York Magazine*. Retrieved 2024, from https://nymag.com/intelligencer/article/ai-artificial-intelligence-chatbots-emily-m-bender.html.
- Armstrong, K. 2023. "ChatGPT: US lawyer admits using AI for case research." BBC. Retrieved 2024, from https://www.bbc.com/news/world-us-canada-65735769.
- NIST Joint Task Force. 2020. NIST SP 80-53 Security and Privacy Controls for Information Systems and Organizations. Revision 5.
- (Microsoft) CoPilot. 2025. [Online]. Available: https://copilot.microsoft.com/. [Accessed June 2024].
- (Microsoft) CoPilot. 2025. [Online]. Available: https://copilot.microsoft.com/. [Accessed February 2025].
- Abrams, M., and J. Weiss. 2008. "Malicious Control System Cyber Security Case Study — Maroochy Water Services, Australia." MITRE Technical Report.
- (OpenAI) ChatGPT. 2023. [Online]. Available: https://chat.openai.com/. [Accessed October 2023].
- Slay, J., and M. Miller. 2008. "Chapter 6: Lessons Learned from the Maroochy Water Breach." In IFIP International Federation for Information Processing 253, Critical Infrastructure Protection, Boston, US-MA: Springer, pp. 73-82.
- Camper, E. 1993. "The Honey Pot: A Lesson in Creativity and Diversity." Retrieved 2025 from https://www.insulators.info/articles/ppl.htm.
- Hassell, S., B. Wilson, and P. Williams. 2020. "Cyber Secure and Resilient Techniques for Architecture." IEEE/NDIA/INCOSE Systems Security Symposium, April.

ABOUT THE AUTHOR

Dr. Beth Wilson is an INCOSE Expert Systems Engineering Professional (ESEP) and INCOSE Systems Security Working Group co-chair. She retired from Raytheon after 33 years as a systems engineer and is currently an adjunct professor at Worcester Polytechnic Institute in their graduate systems engineering program.

Systems Engineering: The Journal of The International Council on Systems Engineering

Call for Papers

The *Systems Engineering* journal is intended to be a primary source of multidisciplinary information for the systems engineering and management of products and services, and processes of all types. Systems engineering activities involve the technologies and system management approaches needed for

- definition of systems, including identification of user requirements and technological specifications;
- development of systems, including conceptual architectures, tradeoff of design concepts, configuration management during system development, integration of new systems with legacy systems, integrated product and process development; and
- deployment of systems, including operational test and evaluation, maintenance over an extended life-cycle, and re-engineering.

Systems Engineering is the archival journal of, and exists to serve the following objectives of, the International Council on Systems Engineering (INCOSE):

- To provide a focal point for dissemination of systems engineering knowledge
- To promote collaboration in systems engineering education and research
- To encourage and assure establishment of professional standards for integrity in the practice of systems engineering
- To improve the professional status of all those engaged in the practice of systems engineering
- To encourage governmental and industrial support for research and educational programs that will improve the systems engineering process and its practice

The journal supports these goals by providing a continuing, respected publication of peer-reviewed results from research and development in the area of systems engineering. Systems engineering is defined broadly in this context as an interdisciplinary approach and means to enable the realization of successful systems that are of high quality, cost-effective, and trustworthy in meeting customer requirements.

The *Systems Engineering* journal is dedicated to all aspects of the engineering of systems: technical, management, economic, and social. It focuses on the life-cycle processes needed to create trustworthy and high-quality systems. It will also emphasize the systems management efforts needed to define, develop, and deploy trustworthy and high quality processes for the production of systems. Within this, *Systems Engineering* is especially concerned with evaluation of the efficiency and effectiveness of systems management, technical direction, and integration of systems. *Systems Engineering* is also very concerned with the engineering of systems that support sustainable development. Modern systems, including both products and services, are often very knowledge-intensive, and are found in both the public and private sectors. The journal emphasizes strategic and program management of these, and the information and knowledge base for knowledge principles, knowledge practices, and knowledge perspectives for the engineering of systems. Definitive case studies involving systems engineering practice are especially welcome.

The journal is a primary source of information for the systems engineering of products and services that are generally large in scale, scope, and complexity. *Systems Engineering* will be especially concerned with process- or product-line–related efforts needed to produce products that are trustworthy and of high quality, and that are cost effective in meeting user needs. A major component of this is system cost and operational effectiveness determination, and the development of processes that ensure that products are cost effective. This requires the integration of a number of engineering disciplines necessary for the definition, development, and deployment of complex systems. It also requires attention to the lifecycle process used to produce systems, and the integration of systems, including legacy systems, at various architectural levels. In addition, appropriate systems management of information and knowledge across technologies, organizations, and environments is also needed to insure a sustainable world.

The journal will accept and review submissions in English from any author, in any global locality, whether or not the author is an INCOSE member. A body of international peers will review all submissions, and the reviewers will suggest potential revisions to the author, with the intent to achieve published papers that

- relate to the field of systems engineering;
- represent new, previously unpublished work;
- advance the state of knowledge of the field; and
- conform to a high standard of scholarly presentation.

Editorial selection of works for publication will be made based on content, without regard to the stature of the authors. Selections will include a wide variety of international works, recognizing and supporting the essential breadth and universality of the field. Final selection of papers for publication, and the form of publication, shall rest with the editor.

Submission of quality papers for review is strongly encouraged. The review process is estimated to take three months, occasionally longer for hard-copy manuscript.

Systems Engineering operates an online submission and peer review system that allows authors to submit articles online and track their progress, throughout the peer-review process, via a web interface. All papers submitted to *Systems Engineering*, including revisions or resubmissions of prior manuscripts, must be made through the online system. Contributions sent through regular mail on paper or emails with attachments will not be reviewed or acknowledged.

All manuscripts must be submitted online to *Systems Engineering* at ScholarOne Manuscripts, located at:

https://mc.manuscriptcentral.com/SYS

Full instructions and support are available on the site, and a user ID and password can be obtained on the first visit.

INCOSE

International Council on Systems Engineering

A better world through a systems approach

PRESENTS

FuSE

Future of Systems Engineering

ITS MISSION

FuSE refines and evolves the SE Vision 2035 across competencies, research, tools & environment, practices, and applications

FuSE identifies critical gaps towards the vision realization and initiates & supports relevant actions

FuSE fosters involvement and collaboration within and outside of INCOSE

FuSE educates, shares success, and expands

CONNECT & GET INVOLVED

INCOSE.ORG/FUSE

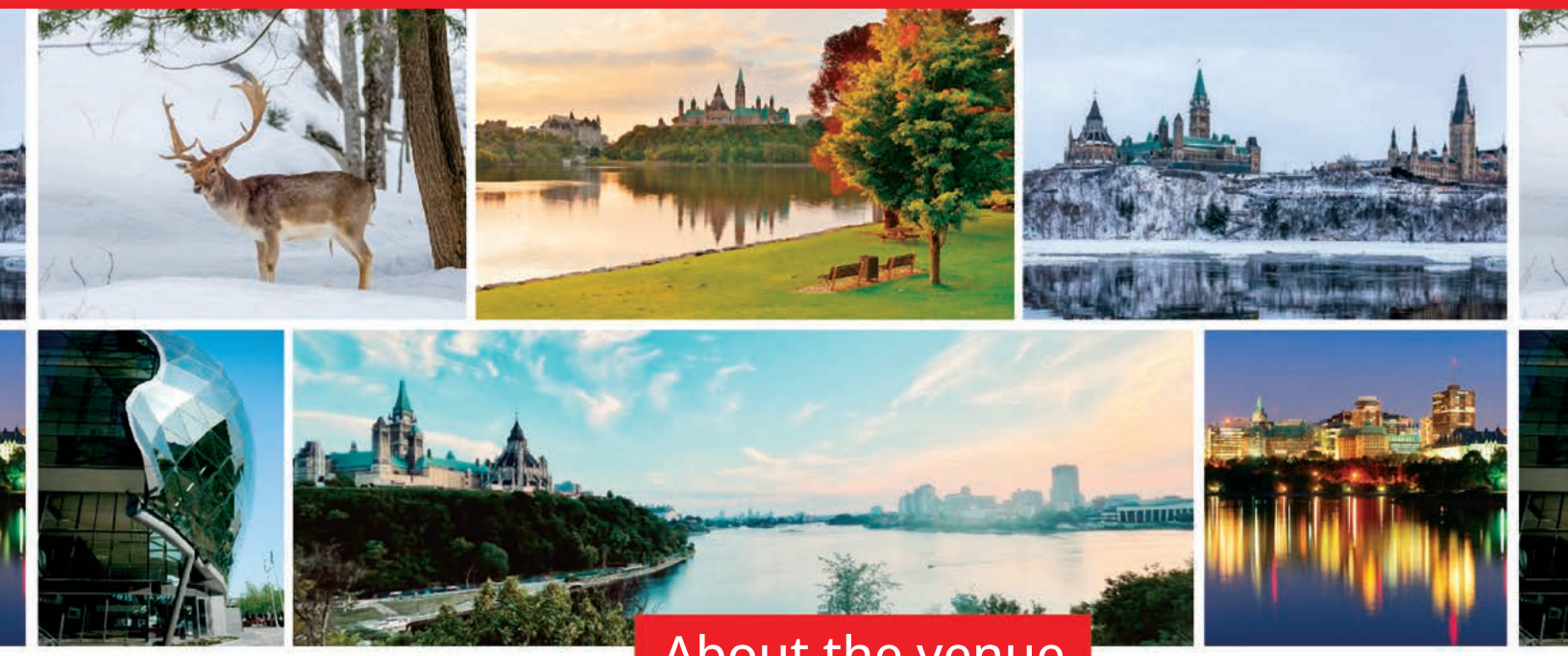
Support by:

DASSAULT SYSTEMES

35th Annual INCOSE international symposium

hybrid event

Ottawa, Canada
July 26 –31, 2025



About the venue

Rogers Centre Ottawa
55 Colonel By Drive
Ottawa, Ontario, Canada

About Rogers Centre Ottawa

Situated on the Old Congress Centre site, the **Rogers Centre Ottawa** breathes new life into Ottawa's downtown core. A striking redevelopment in 2011 tripled the Rogers Centre Ottawa's capacity, drawing attention from national and international planners and making it one of the most successful convention centres in the country.

Getting to Rogers Centre Ottawa

The Rogers Centre Ottawa is conveniently located just a short distance from Ottawa International Airport. Here are your options for getting there:

- **Taxi** or **Ride-Sharing**: This is the most convenient option, offering a direct and hassle-free journey to Rogers Centre Ottawa. Simply book a taxi or use a ride-sharing app like Uber or Lyft upon arrival at the airport. Travel time from the airport is approximately 20 minutes.
- **Public Transportation**: For a more budget-friendly option, you can take the airport shuttle to the city center, then transfer to a local bus or train that stops near the Roger Centre Ottawa. Check the Ottawa Transit Commision (OC Transpo) website for specific routes and schedules.